Technical Report 1022

Analysis and Control of Robot Manipulators with Kinematic Redundancy

Pyung H. Chang

MIT Artificial Intelligence Laboratory

This blank page was inserted to preserve pagination.

ANALYSIS AND CONTROL OF ROBOT MANIPULATORS WITH KINEMATIC REDUNDANCY

BY

PYUNG HUN CHANG

B.S.(1974) Mechanical Engineering
Seoul National University, Seoul, Korea
M.S.(1977) Mechanical Engineering
Seoul National University, Seoul, Korea

Submitted to the Department of Mechanical Engineering
in partial fulfillment of
the requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

at the

Massachusetts Institute of Technology

May 1987

Signature of Author _____

Department of Mechanical Engineering
May 12, 1987

Certified by _____

Berthold K. P. Horn
Department of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by _____

Ain A. Sonin, Chairman
Departmental Graduate Committee
Department of Mechanical Engineering

Analysis And Control Of Robot Manipulators With Kinematic Redundancy

by

PYUNG HUN CHANG

Submitted to the Department of Mechanical Engineering
on May 1, 1987 in partial fulfillment of
the requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

ABSTRACT

A closed-form solution formula for the kinematic control of manipulators with redundancy is derived, using the Lagrangian multiplier method. Differential relationship equivalent to the Resolved Motion Method has been also derived. The proposed method is proved to provide with the exact equilibrium state for the Resolved Motion Method. This exactness in the proposed method fixes the repeatability problem in the Resolved Motion Method, and establishes a fixed transformation from workspace to the joint space. Also the method, owing to the exactness, is demonstrated to give more accurate trajectories than the Resolved Motion Method.

In addition, a new performance measure for redundancy control has been developed. This measure, if used with kinematic control methods, helps achieve dexterous movements including singularity avoidance. Compared to other measures such as the manipulability measure and the condition number, this measure tends to give superior performances in terms of preserving the repeatability property and providing with smoother joint velocity trajectories.

Using the fixed transformation property, Taylor's *Bounded Deviation Paths* Algorithm has been extended to the redundant manipulators.

Thesis Supervisor: Dr. Berthold Klaus Paul Horn

Professor of Electrical Engineering and Computer Science

Acknowledgment: This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Office of Naval Research University Research Initiative Program under Office of Naval Research contract N00014-86-K-0685, in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505, and in part by the Systems Development Foundation.

Acknowledgements

Many people contributed to this thesis. I will attempt to recall the major contributions here.

My advisor Berthold K. P. Horn was the guiding force behind the thesis, whom I deeply thank. He first suggested to me to work on the redundant manipulators, provided advices on the direction of the thesis, and then helped with insights, encouraged, and generously supported me throughout my thesis work at the MIT Artificial Intelligence Laboratory. He also read through several drafts carefully, returning valuable comments.

I also thank the other members of my committee: Jean J. Slotine and Ken Salisbury. Jean has been helping me in various ways for several years. He especially gave me some important insights about sliding mode control techniques. I also appreciate his effort and understanding as the chairman of the committee, which were of great help to me. I would like to thank Ken sincerely for his help in many ways. Out of a genuine interest in my research, he gave me many helpful suggestions in solving some of the problems that I had faced. Above all, his willingness to be available without reservation, together with friendliness, has been especially consoling and encouraging during the period when my advisor was on sabbatical leave.

I also thank Tomás Lozano-Pérez for his excellent teaching through the course in robot manipulation. Besides, his comments and suggestions in the first half of my thesis were valuable. In addition, I recall with thanks the kindness and favor Tomás showed to me on some occasions.

I would like to thank the MIT AI Lab, directed by Patrick Winston, for providing the excellent environment, abundant resources, and generous financial support, which made this thesis work much easier. Additional thanks go to Chae

An for providing me various helps including prayers; Neil Singer for helping me in computation problems including advices on plottings.

Members of the Gate Bible Study at MIT have supported me by their loving care and prayer. My parents and parents-in-law have supported me and encouraged me throughout the seven years at MIT. Now, I cannot recall my two great sons, Chai and Jun, without appreciating and being proud of what they have been and done to me: so disciplined and mature for their ages.

Finally I can never thank enough my wife, Minja, who has demonstrated her love by supporting me both spiritually and emotionally, with such remarkable patience and sensitivity.

Contents

Chapter 1

Introduction

The kinematic control of kinematically redundant manipulators has become an important subject of study, owing to the growing interests in redundant robot manipulators. Unfortunately, most of the existing control algorithms are in a differential form based on the pseudoinverse matrix, subject to usual problems resulting from linearization. The algorithms that instead use a direct mapping, are not compact and closed-form, and not applicable to all manipulators. Furthermore, little is known about the relationship between direct mapping methods and the differential approach for redundant manipulators. One motivation of this thesis is the desire to develop such a compact yet general formula that enables direct mapping, and to understand the relationship between this formula and algorithms in differential form.

Another motivation comes from the observation that the degree of redundancy *cannot* sufficiently describe how far a manipulator is from singularity. In other words, with the same degree of redundancy, there are relatively different degrees of distance from singularity. Surprisingly, little research, to our knowledge, has dealt with this fact. Hence, to develop a satisfactory concept of relative degrees of redundancy is another major purpose. Then, from this concept, we derive

a practical performance measure that can be used for singularity avoidance or dexterity achievement in the redundant manipulators.

The introduction is organized as follows. In Section 1.1, we present the background of the motivations mentioned above. In Section 1.2, then, we define the problems and objectives of this thesis. Finally, in Section 1.3, we present the overview of the organization of this thesis.

1.1 Background Study

1.1.1 Inverse Kinematics For Trajectory Control

Manipulator Kinematics

Manipulator kinematics is a study of manipulator arm motions, from which one finds the relationship between the movement in the workspace and the movement in the joint space. As illustrated in Figure 1.1, the importance of kinematics in

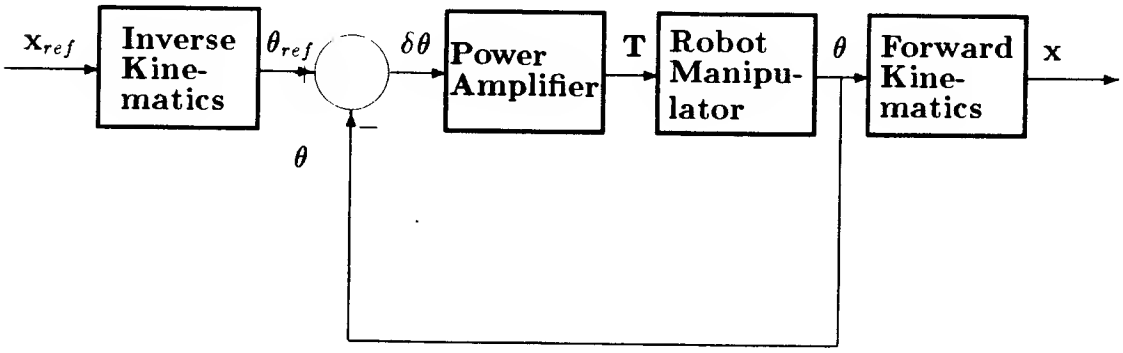


Figure 1.1: The Schematic Diagram Of The Manipulator System.

manipulator trajectory control is clear. It is in workspace coordinates that specific

tasks we desire are usually expressed; whereas it is in the joint coordinates that the actuator movements are described. Therefore, kinematics is a fundamental tool directly connected to the performance in manipulator trajectory control.

Forward Kinematics and Inverse Kinematics

Manipulator kinematics, hence, includes two important problems. One is to find the location[1] of the manipulator in workspace coordinates from its joint coordinates and the other is to find the joint coordinates from the location in the workspace coordinates. The former is called the *forward kinematic problem*; the latter, being the inverse problem, is called the *inverse kinematic problem.*

Of the two problems, the forward kinematic problem is the simpler one since a set of joint coordinates unambiguously determines a unique location in the workspace. Furthermore, the solution can usually be expressed in a symbolic form, which can be easily evaluated given a set of joint coordinates (Paul,1981). The inverse kinematic problem, on the other hand, is more complicated because, given a location in the workspace, there are multiple sets of corresponding joint coordinates. More importantly, there do not exist, in general, closed-form solutions, except for some kinematic structures of manipulators. Pieper(1968), by the way, presented the criteria that guarantee the existence of a closed-form solution. Thus, if a manipulator structure does not belong to this category, one has to find solutions by numerical methods, typically iterative ones.

Resolved Motion Method and Inverse Kinematic Method for Non-Redundant Manipulators

To numerically solve the inverse kinematic problem, there are two approaches: one that uses the linear relationship between the differential joint displacement and

[1] location consists of position and orientation.

differential end effector displacement; and one that uses the direct mapping from workspace to joint space. The former, originally proposed by Whitney(1969,1972) and subsequently used by(Paul,1981; Featherstone,1983), is called the *Resolved Motion Method*; while the latter, investigated by (Albala,1979; Konstantinov et al,1982; Gupta et al,1985; Goldenberg et al,1985; Angeles,1985; Wampler,1986), is simply called the *Inverse Kinematic Method.*

More specifically, the Resolved Motion Method first differentiates the desired trajectory of the end effector to obtain velocity. Then, this Cartesian velocity is mapped into the joint velocity, using the inverse of the Jacobian matrix. Finally, the joint velocity is integrated to determine the joint displacement.

The iterative Inverse Kinematic Method, on the other hand, directly solves the nonlinear kinematic equations, without linearizing them, for a given location in workspace. Note, however, that the updating process at each iteration step of a numerical method for solving the nonlinear kinematic equations can be viewed as an incremental process like the Resolved Motion Method. With this view, then, the essential difference between the two approaches may be considered the number of iterations: The Resolved Motion Method can be considered a *one iteration* Inverse Kinematic Method without any built-in convergence criteria.

Comparison of the two methods

The Resolved Motion Method, thus, has some weak points:

- The method has intrinsic inaccuracy because of the linear approximation characteristics of the Jacobian matrix; thus it accumulates errors, which become larger as the velocity increases.

- The method, being a rate equation, is not self-starting: given a location of the end effector, corresponding joint values cannot be determined without using other methods.

The Inverse Kinematic Method without closed-form solution, on the other hand, usually requires more computational effort than the other method. So this method is not efficient when high accuracy is not needed. Even when high accuracy is required, in order to be practical for the real-time control purpose, the method may need some additional numerical schemes such as interpolations between knot points (Paul,1975; Paul,1979; Taylor,1979). Moreover, for some applications that require to resolve joint velocity and acceleration as well, it has an intrinsic disadvantage. Yet, the Inverse Kinematic Method is still attractive because of the direct mapping from the workspace to joint space, fixing the aforementioned problems of the other method.

1.1.2 Kinematically Redundant Manipulators

Kinematic Redundancy

A *kinematically redundant* robot manipulator is a manipulator that has more degrees of freedom than necessary to place the end effector at a desired location. For example, if we want to place the end effector in a three dimensional-space, we need six degrees of freedom: three for translation and three for orientation. Thus, a robot manipulator with more than six degrees of freedom is kinematically redundant in the three-dimensional space.

Use of Redundancy

The major advantages of adding redundant degrees of freedom to a robot manipulator are as follows:

1. One achieves greater dexterity in maneuvering in a workspace with obstacles.

2. One can avoid singular configurations of the manipulators.

Because of these significant advantages, an increasing amount of research has focused on the kinematically redundant manipulator, and the progress in this field has been rapid. More specifically, to avoid obstacles, several researcher such as (Yoshikawa,1984; Maciejewski,1985; Espiau,1985; Nakamura,1985; Baillieul,1986) have used the kinematical redundancy. Meanwhile the effort to avoid singularity through the use of redundant manipulators is found in the works by (Whitney,1972; Yoshikawa,1984; Hollerbach,1985,Baillieul,1985; Luh et al,1985; Nakamura,1985a). Besides, the use of the kinematic redundancy has been proposed for constraining the joint variables within their physical limits (Liegeois,1977), or for minimizing joint torques (Hollerbach,1985), or for minimizing the kinetic energy due to joint velocity (Whitney,1972).

Redundancy control

Mathematically, the inverse kinematic problems of redundant manipulators are under-determined problems: more variables(joint variables) than constraints(kinematic equations). In order to solve equations we need to impose extra constraints. These additional constraints sometimes tend to be imposed out of necessity to fully specify the under-determined condition; or sometimes are *used*, on purpose, to achieve additional performances objectives as mentioned above. The former tendency was shown in (Whitney,1972), where the joint velocity was obtained by using the pseudoinverse of the Jacobian matrix to resolve the under-determined joint velocities.

The latter case, active use of redundancy called the redundancy control, was first proposed by Liegeois (Liegeois,1977). He used, in addition to the same pseudoinverse term as above, the null space term, where he included the gradient vector of a scalar function that represents the desired performance. This gradient vector by the way, if used with the null space of the Jacobian matrix, forces

the joints to move toward the direction where the scalar function has the optimal value at that instant. This formulation of solution, owing to the ease of including the desired performance, has been rather extensively used to achieve various performances as mentioned above (Klein and Huang,1983; Maciejewski and Klein,1985; Nakamura,1985a; Hollerbach and Suh,1986).

Resolved Motion Method and Inverse Kinematic Method

In much of the research just mentioned — regardless of whether the null space is actively used or not, or regardless of which performance is desired — the motion is resolved in the differential form. More specifically, the differential displacement (or velocity) of the end effector is mapped into the differential joint displacement (or velocity) now by using the pseudoinverse of the Jacobian matrix, and then incrementally determines the joint displacement. This technique, thus, is essentially the same as the Resolved Motion Method in the nonredundant case; the only difference is that the pseudoinverse is used instead of the inverse of the Jacobian matrix.[2]

In contrast to this direction of research, relatively little research for kinematically redundant manipulators, e.g., (Benati et al,1982; Hollerbach,1985; Oh et al,1984; Benhabib et al,1984,1985) has involved the direct mapping — the counterpart of the Inverse Kinematic Method in the nonredundant case. In the redundant case too, there exist in general no closed-form solutions; only if certain conditions are met by the manipulator structure, then a part of solution is given. For example, in (Benati et al,1982, Hollerbach, 1985) only some of the joint variables were obtained symbolically. To obtain these solutions the manipulator structure and the number of degrees of freedom were fixed, explicitly in (Hollerbach,1985) and implicitly in (Benati,1982).

[2]Hence we will call this resolution technique the Resolved Motion Method, too.

Comparison of the two methods in the redundant case

Of the two methods, the Resolved Motion Method and the Inverse Kinematic Method, the advantages(and disadvantages) of the one method over the other in the nonredundant case were compared in the previous subsection. The comparison still holds in the redundant case, since each of the methods is essentially the same as its counterpart in the nonredundant case.

In addition, another significant drawback in the Resolved Motion Method was pointed out in (Klein,1983): the lack of repeatability, the ability to repeat the same joint values for repeated end effector motion. This problem, however, does not occur when direct mapping is used.

Meanwhile, an additional difficult task for Inverse Kinematic Method, on the other hand, is to rationally (or optimally) use the extra degrees of freedom when achieving the additional objectives. Often, this optimization procedure being rather complicated, the overall inverse kinematic process results in a series of iterative procedures. This time, the shortcoming is not so serious in the Resolved Motion Method, because of the simple null space expression.

Therefore, the comparison clearly shows that one method is complementary to the other, indicating one desired direction in which a kinematic control formula should be developed: a formula that provides with the direct mapping, and that is as concise and general as the Resolved Motion Method.

1.1.3 Performance Measure For Singularity Avoidance

Using a quantitative measure that represents the desired performance is a beneficial method for the analysis, design, and control of engineering systems, as follows:

- one can evaluate the performance of a given system and analyze the system, by estimating this measure; or
- one can design a system that achieves the performance in a certain degree, by maximizing(or minimizing) this measure; or
- one can control, on the on-line basis, a given system to achieve it, by maximizing the measure at each moment,

without having to rely solely on experience and intuition.

In the robotic system, also, various performance measures have been incorporated to quantify desired performance features listed in the previous subsection.

Dexterity measure

One of these performances was the ability to avoid singularity, or in a broader sense the ability to dexterously move the end effector to an arbitrary location within the workspace, without getting into singular configurations. To quantitatively represent the ability, which may be called dexterity, several performance measures have been proposed (Yoshikawa,1985a,1985b; Uchiyama,1985; Maciejewski,1985; Salisbury; 1982). These measures are the determinant, the condition number, and a few combinations of singular values, of the Jacobian matrix $\mathbf{J}$ — through which, by the way, the end effector movement is achieved.

Determinant

In linear algebra, the determinant of a matrix has been an important measure used to test the invertibility of the matrix and its nearness to singularity. Accordingly the determinant of the Jacobian matrix has been tried for the dexterity measure for both nonredundant and redundant manipulators. For nonredundant manipulators, for instance, the determinant has been used as a measure of degeneracy for the analysis of the wrist configurations(Paul and Stevenson, 1983).

For redundant manipulators, on the other hand, Yoshikawa(1984) has proposed a measure called *manipulability*, defined as the square root of the determinant of $\mathbf{J}\mathbf{J}^T$. This measure is viewed as a generalized concept of the determinant, because of the followings:

- the manipulability reduces to the regular determinant in the nonredundant case.
- the manipulability become zero, when workspace rank reduces at singularity, just as the regular determinant of a square Jacobian matrix does.
- since the singular values of $\mathbf{J}\mathbf{J}^T$ have the square values of those of $\mathbf{J}$, the determinant of $\mathbf{J}\mathbf{J}^T$ may be regarded as if it were the square of the regular determinant of a square Jacobian matrix.

Condition number

Meanwhile, since the condition number of the Jacobian matrix is another important measure that also indicates the nearness of a matrix to singularity, it has been proposed for a dexterity measure(Salisbury,1982). It is noteworthy that this measure was initially used to determine the configuration that minimizes the propagation from the torque error to the force error — equivalently, the velocity error propagation from joint space to workspace — for nonredundant manipulator.

Singular values

The determinant and the condition number can be also expressed in terms of singular values of the Jacobian matrix: the former is the product of all the singular values, the latter the ratio of the largest to the smallest singular value. Since the minimum singular value becomes zero when the matrix is singular, and approximately determines the worst limits of the two measures, the value itself was suggested as a new measure(Klein,1985). In addition to its simple expression,

the measure has a relatively clear physical meaning: it may be interpreted as the minimum responsiveness in end effector velocity due to a unit change in joint velocity(Klein,1985).

Besides, the geometric mean and harmonic mean of singular values have been proposed for the dexterity measures(Yoshikawa,1985b), which may be viewed essentially as variations of aforementioned measures.

Common features

The features common to all these measures are as the following:

- They indicate the presence of singularity: when singular, the value of these measures become zero, except for the condition number, the value of which becomes infinity.

- Their absolute values — inverse of the value in the case of the condition number — appear to represent, in one way or another, the farness or distance from singularity. That is, the larger the value, the farther is the manipulator from a singularity.

In the case of redundant manipulators, however, these measures do not explicitly indicate the successive changes in the available degrees of freedom as long as the workspace rank is preserved. For instance, suppose we have a five d.o.f. manipulator which is to move in a three-dimensional workspace, hence having two degrees of redundancy. Although the manipulator happens to lose one degree of freedom, or even two, the measures do not necessarily indicate that fact.

Problems when losing d.o.f.

Losing degrees of freedom may not in itself be a serious drawback, as long as the workspace rank is fully preserved so that the desired location of the end effector can be achieved by joint variables. Yet, what may be of more concerns

are potential problems that are expected to arise — from the similar experience in the nonredundant case — when degrees of freedom are lost. More specifically speaking, in the nonredundant case, the point where the degrees of freedom are lost — namely the singular point — is in fact the boundary of switching from one set of joint solution to another (Uchiyama,1979). Once that switching happens, the manipulator tends to *stay* in the new kind of joint configuration different from the previous kind, thus causing another pattern of repeatability problem. Besides, when the switching arises, usually there are accompanying discontinuity in motion, resulting in large joint velocities. The same problems are expected in the redundant case, since in this case too there exist multiple solutions of different kinds (Borrel,1986) whose boundaries are the points where the degree of freedom decreases. It appears, however, that these nontrivial problems tend to be veiled because of the fact that owing to the redundancy the switching can happen without causing the more serious problem, singularity. To our knowledge, there have not appeared any analysis on these problems for redundant manipulators, and any performance measures that are intended to prevent them.

1.2 Objectives

The objective of this thesis, broadly speaking, is to study trajectory control of kinematically redundant manipulators, focusing on the kinematic problems mentioned in the first subsection.

To this end there are two major goals. The first goal is to develop a general closed-form method for the inverse kinematics of manipulators with kinematic redundancy. This method is similar to the formulation of the Resolved Motion method, in actively using kinematic redundancy; but it is different in choosing direct mapping from the workspace to joint space. Under the first goal, however,

to understand the relationship between the inverse kinematic method and resolved motion method is another intended purpose.

The second goal is to analyze the aforementioned relative distance of redundant manipulators, and to derive from this analysis a performance measure that represents the dexterity of manipulator including singularity overcoming. This performance measure is intended to be used either with on-line kinematic control methods including the one developed from the first goal, or for off-line design purposes. Besides, the relationship between the resulting measure and already existing measures is to be examined.

1.3 Overview of the Thesis

In Chapter 2, a closed-form formula for inverse kinematics of kinematically redundant manipulators is to be derived using the Lagrangian multiplier method. This formula consists of a set of equations which, if added to the kinematic equations, fully constrain the initially under-determined problem. The mathematical meaning and applications of the formula are examined. Then the relationship between the formula and already existing similar methods is investigated.

In Chapter 3, the qualitative relationship between the new method and the Resolved Motion Method is examined. Then in the light of the relationship, the *repeatability* problem is focused on.

In Chapter 4, the qualitative results obtained in Chapters 2 and 3 are verified through numerical experiments. Finally numerical efficiencies of the two methods are compared.

In Chapter 5, we propose a new concept representing the distance from singularity for kinematically redundant case. This concept is obtained by observing the structure of the Jacobian matrix of redundant manipulators. Then, by using

the concept, we derive a new dexterity measure that can be used with either the the formula obtained in Chapter 2 or the Resolved Motion Method.

In Chapter 6, the new performance measure is compared with two existing performance measures: the manipulability measure and the condition number. After their qualitative relationships are examined, the numerical experiments are made with redundant manipulators to compare the effectiveness of each measure in achieving dexterous movements. In the comparison, at the same time, the repeatability problem, as well as the ability to preserve the *kind* of joint solutions are to be observed.

Finally, in Chapter 7, concluding remarks are to be made.

Chapter 2

The Proposed Method for the Kinematic Control

2.1 Introduction

In this chapter, we derive a general closed-form formula for the kinematic control of redundant manipulators by using the Lagrangian multiplier method. This formula, more specifically speaking, consists of an additional set of constraints beside the kinematic equation.

The key features of this formula, beside its conciseness, are that it can contain the performance measure as easily and generally as the Resolved Motion Method with null space can (Liegeois,1977); and at the same time it is an *inverse kinematic method.* Among inverse kinematic methods for redundant manipulators, we find similar approaches that use performance measures; but they lack some of the features this formula has. Benati et al (Benati,1982) suggested a method using a general quadratic performance measure to be optimized, the resulting control algorithm of which is quite complicated because of including the measure. Benhabib et al (Benhabib,1984,1985) also used performance measures to achieve

additional performances. These measures are optimized with a searching algorithm under constraints on joint variables. The resulting algorithm becomes also complicated.[1]

Rather, among the approaches using the Resolved motion method, we find additional constraints in similarly concise forms, which could be also used in the inverse kinematic method. Baillieul(Baillieul,1985), in the process of deriving a rate equation called the *extended Jacobian Method*, obtained null space expressions similar to the formula proposed here. Luh and Gu(Luh,1985) proposed another expression of the null space in a generic form. We will investigate more in detail about the relationship between the proposed formula and these methods.

After deriving the formula in Section 2.2, we will examine in Section 2.3 its mathematical meaning and characteristics, as well as computational aspects in Section 2.4. Then the relationship between the formula and the aforementioned methods is covered in Section 2.5, and finally some concluding remarks are made in Section 2.6.

2.2 Derivation of the Proposed Equation

In this section, we will derive extra equations which, together with the kinematic equations of the manipulator, can fully specify the under-determined problem.

The kinematic equation for the redundant manipulator is given as the following vector equation:

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) \tag{2.1}$$

where $\mathbf{x}$ is an m-dimensional vector representing the location of the end effector with respect to the base coordinate system in the workspace, $\boldsymbol{\theta}$ is n-dimensional vector representing joint variables, and $\mathbf{f}$ is a vector function consisting of m

[1] If there were no constraints on joint variables, the algorithm would be simpler.

scalar functions, with $m < n$. (2.1) may be rewritten as

$$\begin{aligned} \mathbf{F}(\boldsymbol{\theta}) &= \mathbf{f}(\boldsymbol{\theta}) - \mathbf{x} \\ &= \mathbf{0} \end{aligned} \tag{2.2}$$

Let $H(\boldsymbol{\theta})$ be the aforementioned criteria function(or performance measure) with continuous first-order partial derivatives with respect to joint variables, which quantitatively represents the desired performance.

Let us define the Lagrangian function $L(\boldsymbol{\theta})$ as the following:

$$L(\boldsymbol{\theta}) = \boldsymbol{\lambda}^T \mathbf{F}(\boldsymbol{\theta}) + H(\boldsymbol{\theta}) \tag{2.3}$$

where $\boldsymbol{\lambda}$ is an m-dimensional Lagrangian multiplier vector. At the stationary points of L,

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{\theta}} &= \boldsymbol{\lambda}^T \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}} + \frac{\partial H}{\partial \boldsymbol{\theta}} \\ &= \mathbf{0} \end{aligned} \tag{2.4}$$

where the $m \times n$ matrix, $\frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}}$, is the Jacobian matrix $\mathbf{J}$ (Whitney,1972). In (2.4), as $\mathbf{x}$ is expressed in the base coordinate system, so are the Lagrangian function and the Jacobian matrix. The second term in the right hand side(r.h.s.) of (2.4) is the transpose of the gradient vector $\mathbf{h}$ such as

$$\begin{aligned} \mathbf{h} &= (h_1, h_2, ..., h_n)^T \\ h_i &= \frac{\partial H}{\partial \theta_i}, \quad (i = 1, 2, .., n) \end{aligned} \tag{2.5}$$

Thus, (2.4) becomes the following:

$$\boldsymbol{\lambda}^T \mathbf{J} = -\mathbf{h}^T$$

Transposing, we get

$$\mathbf{J}^T \boldsymbol{\lambda} = -\mathbf{h}$$

or

$$\begin{bmatrix} (J^1)^T \\ (J^2)^T \\ . \\ . \\ (J^n)^T \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ . \\ . \\ \lambda_m \end{bmatrix} = - \begin{bmatrix} h_1 \\ h_2 \\ . \\ . \\ h_n \end{bmatrix} \tag{2.6}$$

where $(J^i)^T$ denotes the transpose of i-th column vector of the Jacobian matrix. In (2.6), we have n linear equations with m unknowns, $\lambda_1, \lambda_2, .., \lambda_m$. Selecting m linearly independent equations from (2.6), which may be chosen to be, without loss of the generality, the first m equations, we have,

$$\begin{bmatrix} (J^1)^T \\ (J^2)^T \\ . \\ . \\ (J^m)^T \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ . \\ . \\ \lambda_m \end{bmatrix} = - \begin{bmatrix} h_1 \\ h_2 \\ . \\ . \\ h_m \end{bmatrix} \tag{2.7}$$

Inverting, we obtain $\boldsymbol{\lambda}$, as

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ . \\ . \\ \lambda_m \end{bmatrix} = - \begin{bmatrix} (J^1)^T \\ (J^2)^T \\ . \\ . \\ (J^m)^T \end{bmatrix}^{-1} \begin{bmatrix} h_1 \\ h_2 \\ . \\ . \\ h_m \end{bmatrix}$$

Substituting this into the remaining $n - m$ equations in (2.6), we have

$$- \begin{bmatrix} (J^{m+1})^T \\ (J^{m+2})^T \\ . \\ . \\ (J^n)^T \end{bmatrix} \begin{bmatrix} (J^1)^T \\ (J^2)^T \\ . \\ . \\ (J^m)^T \end{bmatrix}^{-1} \begin{bmatrix} h_1 \\ h_2 \\ . \\ . \\ h_m \end{bmatrix} = - \begin{bmatrix} h_{m+1} \\ h_{m+2} \\ . \\ . \\ h_n \end{bmatrix} \tag{2.8}$$

For brevity, let us denote

$$\mathbf{J_m} = \begin{bmatrix} (J^1)^T \\ (J^2)^T \\ . \\ . \\ (J^m)^T \end{bmatrix}; \mathbf{J_{n-m}} = \begin{bmatrix} (J^{m+1})^T \\ (J^{m+2})^T \\ . \\ . \\ (J^n)^T \end{bmatrix}; \quad \mathbf{h_m} = \begin{bmatrix} h_1 \\ h_2 \\ . \\ . \\ h_m \end{bmatrix}; \mathbf{h_{n-m}} = \begin{bmatrix} h_{m+1} \\ h_{m+2} \\ . \\ . \\ h_n \end{bmatrix}$$

Adding $\mathbf{h_{n-m}}$, and multiplying both sides of (2.8) by -1, we have

$$\mathbf{J_{n-m}J_m}^{-1}\mathbf{h_m} - \mathbf{h_{n-m}} = \mathbf{0}$$

which may be alternatively expressed as

$$\left[\mathbf{J_{n-m}J_m}^{-1} : -\mathbf{I_{n-m}}\right] \begin{bmatrix} \mathbf{h_m} \\ \mathbf{h_{n-m}} \end{bmatrix} = \mathbf{0}$$

where $\mathbf{I_{n-m}}$ is an identity matrix of rank $(n-m)$. If we denote

$$\mathbf{Z} = \left[\mathbf{J_{n-m}J_m}^{-1} : -\mathbf{I_{n-m}}\right] \tag{2.9}$$

Then (2.8) becomes

$$\mathbf{Zh} = \mathbf{0} \tag{2.10}$$

where $\mathbf{Z}$ and $\mathbf{h}$ are defined as above. If we combine the kinematic equation, (2.1), with (2.10), as a system of equations, we get

$$\left\{ \begin{array}{ll} \mathbf{x} & = \mathbf{f}(\boldsymbol{\theta}) \\ \mathbf{Zh} & = \mathbf{0} \end{array} \right\} \tag{2.11}$$

Since $\mathbf{Z}$ is an $(n-m) \times n$ matrix, and $\mathbf{h}$ is an n-dimensional vector, (2.10) consists of $(n-m)$ scalar equations with n unknowns, $\boldsymbol{\theta}$. On the other hand, the kinematic equation, (2.1), has m scalar equations. Therefore, (2.11) has n independent nonlinear equations which now fully specify the n unknowns.

Note that finally we have derived the expected formula, Equation 2.11 — a simple yet general formula that provides with a direct mapping from workspace

to joint space. In (2.11), the additional set of constraints, (2.10), resolve the redundancy — at the inverse kinematic level — in such a way that, under the kinematic constraint of (2.1), the criteria function, $H(\boldsymbol{\theta})$, may be minimized.

Note also that (2.11) has to be solved numerically.

2.3 Characteristics

In this section, we will examine the additional set of constraints, with the kinematic equation, more in depth. The fact that the matrix $\mathbf{Z}$ consists of the elements of $\mathbf{J}$ alone implies that it may have a close relationship with the Jacobian matrix. Then, what is the relationship? The following theorems gives the answer.

Theorem 1 *The rank of* $\mathbf{Z}$ *defined in (2.10) is* $n - m$

Proof:

The rank of any matrix is the dimension of its largest nonsingular submatrix. Since $\mathbf{I_{n-m}}$ is the largest nonsingular matrix for $\mathbf{Z}$, the rank of $\mathbf{Z}$ (and $\mathbf{Z}^T$, too) is $n - m$.

Theorem 2 $\mathbf{JZ}^T = \mathbf{0}$

Proof:

$$\begin{aligned} \mathbf{JZ}^T &= [\mathbf{J_m}^T : \mathbf{J_{n-m}}^T] \begin{bmatrix} (\mathbf{J_m}^{-1})^T (\mathbf{J_{n-m}}^{-1})^T \\ -\mathbf{I_{n-m}} \end{bmatrix} \\ &= \mathbf{0} \end{aligned} \tag{2.12}$$

Here, Theorem 2 shows that all the column vectors of $\mathbf{Z}^T$ are orthogonal to $\mathbf{J}$, whereas Theorem 1 tells us that the number of the column vectors are equal to the rank. Thus, column vectors of $\mathbf{Z}^T$(or row vectors of $\mathbf{Z}$) are a set of *basis* vectors which span the null space of $\mathbf{J}$. From this fact, we may conjecture that Equation

(2.10) could be the direct counterpart of the homogeneous solution term of the Resolved Motion Method, which also uses the null space to resolve the redundancy. This conjecture will be proved in Chapter 3, where the relationship between the proposed formula and the Resolved Motion Method will be focused on more in detail.

In (2.10), note that the matrix **Z** depends on the intrinsic kinematic property of a manipulator, while **h** depends on an arbitrarily imposed property. Therefore, if a symbolic form of **Z** is available — which is not very difficult, once the Jacobian matrix can be expressed in a symbolic form — we have only to replace **h**, depending on the desired performance, without having to derive the equation all over again. For example, consider a seven degrees of freedom robot having the following Jacobian matrix as shown in (Luh,1985):

$$\begin{bmatrix} -a_2c_{23} & l_2s_3+a_3 & a_3 & 0 & 0 & 0 & 0 \\ a_3s_{23}+l_2c_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -a_3s_{23} & -l_2c_3 & 0 & 1 & 0 & 0 & 0 \\ -s_{23} & 0 & 0 & 0 & 0 & -s_4 & c_4s_5 \\ 0 & 1 & 1 & 0 & 0 & c_4 & s_4s_5 \\ c_{23} & 0 & 0 & 0 & 1 & 0 & c_5 \end{bmatrix}$$

where l_2, l_3, and a_2. a_3 represent the link parameters, while the variables with subscripts are defined as

$$\begin{aligned} &s_i = sin(\theta_i), \quad s_{ij\ldots k} = sin(\theta_i+\theta_j+\ldots+\theta_k), \\ &c_i = cos(\theta_i), \quad c_{ij\ldots k} = cos(\theta_i+\theta_j+\ldots+\theta_k), \quad i,j,..,k = 1,2,3 \end{aligned} \tag{2.13}$$

Substituting the Jacobian matrix into (2.10), and symbolically manipulating with MACSYMA(Macsyma Manual), we have **Z** in the following form:

$$\mathbf{Z} = [0 \;\; a_3s_5 \;\; (l_2s_3+a_3)s_5 \;\; a_3l_2c_3s_5 \;\; l_2s_3s_4c_5 \;\; l_2s_3c_4s_5 \;\; -l_2s_3s_4]\,(a_3s_{23}+l_2c_2)$$

Note that we have derived an expression of **Z**, apart from **h**.

In (2.10), *any* criteria function may be used, as long as the function can be reduced to an expression in terms of joint variables only. For instance, consider the following criteria function, $H(\boldsymbol{\theta})$, for the obstacle avoidance problem(Nakamura,1985)[2]:

$$H = k_O \sum_{i=1}^{l} 1/\{C_O(\mathbf{x}_i) - 1\} + k_J \sum_{j=1}^{n} 1/(\theta_{jmax}^2 - \theta_j^2), \tag{2.14}$$

where k_O and k_J are scaling factors; $\mathbf{x}_i = (x_{1i}, x_{2i}, x_{3i})^T$ is the position of the i-th point among l points on the manipulator; θ_{jmax} is the limit of j-th joint; and the model of the obstacle in the workspace, $C_O(\mathbf{x}_i)$, is defined as,

$$C_O(\mathbf{x}_i) = \sum_{k=1}^{3} \left(\frac{x_{ki} - x_{kc}}{r_k}\right)^s \tag{2.15}$$

where x_{kc}, r_k, and s are the center coordinate, radii, and roundness exponent of the obstacle object. Note that we can reduce H to a function of $\boldsymbol{\theta}$ only, by transforming $\mathbf{x}_i$ into $\mathbf{f}_i(\boldsymbol{\theta})$ using (2.1), and thus can apply (2.10) to it.

2.4 Computational Consideration

In this section, an analysis will be made with regard to the computational effort required to solve (2.11). Much of the analysis is based on the various results of the research on computational efforts. So one can find more detailed discussions in the references cited in this subsection.

For the sake of generality, we consider the general manipulator without any special geometry, thus assuming that $\mathbf{J}$ and $\mathbf{Z}$ are not given in the symbolic form. A special geometry that allows the symbolic form, of course, would enable much more efficient computations.

The computational effort will be measured in terms of arithmetic operations such as addition, subtraction, multiplication, and division. Furthermore, we assume that the four arithmetic operations require approximately the same amount

[2]the problem was originally treated in the dynamic context by using a potential function.

of computation time, and the effort required to evaluate trigonometric functions is negligible as compared to the total computational effort.

2.4.1 Motion strategy

In order to compute the joint trajectory for a given Cartesian path, We need, briefly speaking, to take the following procedures: we first determine *via* points on the path; then obtain, by using inverse kinematic methods, corresponding points in the joint space; and finally concatenate these angles to generated the joint trajectory.

How to generate the via points, and how to determine an *appropriate* number of the points are important subjects in the motion planning area. Too many via points result in a wasteful computation and slow motion, whereas too few result in inaccurate trajectory control. In addition to deciding the via points, how to generate intermediate points between the via points is another important issue. Usually interpolation methods are used to generate intermediate points, which we will call *knot points.*

When obtaining joint values corresponding to a via point, we use iterative methods if a symbolic solution is not available. So, determining joint angles for a via point involves a number of iteration steps; each step in turn requires a series of computations. Hence, the total computational effort, N_{total}, required for a given trajectory is obtained in general as the following:

$$\begin{aligned} N_{total} &= N_{con} + \sum_{k=1}^{N_{via}} N_k \\ N_k &= N_{iteration} N_{step} \end{aligned}$$

where N_{con} is the computational effort for concatenating knot points either in workspace or joint space, N_{via} the number of via points, N_k the computational effort for inverse kinematics at each point, $N_{iteration}$ the number of iterations at

the point, and N_{step} the computational effort at each iteration step. Assuming that, as compared to the other term in N_{total}, N_{con} is relatively small, we will only discuss each of remaining terms, in a reverse order.

2.4.2 Computational Effort at Each Iteration Step

If (2.11) is to be solved numerically, the computational effort at each iteration step, N_{step}(or N_1 in this case), is evaluated as follows:

$$N_1 = N_{FK} + N_J + N_{Zh} + N_{n.l} \tag{2.16}$$

where N_{FK} denotes the computational effort required for the forward kinematics, (2.1), N_J the effort to obtain the Jacobian matrix, N_{Zh} for **Zh** in (2.10), and $N_{n.l.}$ the effort needed by the numerical method for solving the system of nonlinear equations.

For the general manipulator with revolute joints — no significant difference is expected when a few prismatic joints are included — N_{FK} is given as (Angeles,1985)

$$\begin{aligned} N_{FK} &= N_R + N_T \\ N_R &= 36(n-1) + 4; \quad N_T = 16(n-1) + 2 \end{aligned} \tag{2.17}$$

where N_R and N_T indicate the computational efforts required to compute orientation and position of the end effector[3].

N_J for the general manipulator, with Waldron's scheme, is given as (Orin,1984)

$$N_J = 45n - 93 \tag{2.18}$$

Since we have not, at this point, chosen a specific criteria function from a variety of choices, let us simply assume that we have derived **h** with relatively a small

[3] We derived these general formula on the basis of the computation algorithm in (Angeles,1985). Slightly different formula can result depending on different computation details

amount of computation as compared to the other parts of computation. If we assume at the same time that a symbolic form of **Z** is not available, **Zh** in (2.10) is more efficiently computed by first solving for $\boldsymbol{\lambda}$ in (2.7) with Gaussian elimination, and then substituting it as in (2.8). Thus, N_{Zh} is given as follows:

$$\begin{aligned} N_{Zh} &= N_G + N_s \\ N_G &= \frac{2m^3}{3} + \frac{3m^2}{2} - \frac{7m}{6}; \quad N_s = (n-m)(2m-1) \end{aligned} \tag{2.19}$$

where N_G is the effort for Gaussian elimination(Nakamura,1985), and N_s is that for the substitution.

If we use, for example, MINPACK-1, one of the well-known software packages, to solve the system of nonlinear equations, the computational effort, $N_{n.l}$, becomes (MINPACK Manual)

$$N_{n.l} = 11.5n^2$$

This package, by the way, is primarily based on an improved version of Powell's hybrid method, a method that combines the Newton-Raphson method with the steepest descent method (MINPACK Manual).

Consequently, the total effort of computation for the general manipulator with kinematic redundancy per iteration for the proposed method becomes as follows:

$$N_1 = \frac{2}{3}m^3 + 11.5n^2 + 2nm - \frac{m^2}{2} - \frac{m}{6} + 96n - 139 \tag{2.20}$$

As an example to show the total computational effort as well as the relative significance of each part in it in a realistic situation, let us evaluate the effort when $n = 7$ and $m = 6$. According to aforementioned estimations, we have $N_1 = 1306$, in which, $N_{FK} = 318$, $N_J = 222$, $N_{Zh} = 202$, $N_{n.l} = 564$.

2.4.3 Number of Iterations

The number of iterations required for a system of nonlinear equations depends on several factors. Among these, there are some major ones such as the initial esti-

mate of solution, the particular updating algorithm being used, and the distance of the estimate at the present step from singularities.

In the inverse kinematic context when the end effector is successively passing adjacent points, the initial estimates are usually good enough to use such a fast converging algorithm as the Newton-Raphson method. For applications with poor initial estimates, however, we need some correctional algorithms such as the steepest descent method. Alternatively, we may use a method that combines the Newton-Raphson method and steepest descent method, such as the Powell's hybrid method mentioned in the previous subsection.

With the Powell's method, solving a system of nonlinear equations that consists of trigonometric functions — thus very similar to the kinematic equations — is reported to require from about 12 to 15 iterations for systems of up to *nine* equations(Rabinowitz,1970). This data shows a good agreement with the result by Benhabib et al(1986), which reports that the number of iterations for seven degrees of freedom redundant manipulators is less than 15.

2.4.4 Minimum Number of Via Points

One extreme approach to control the motion of the end effector is to obtain joint solutions at every sample interval. Obviously it requires considerable amount of real time computation. One possible way to improve the approach is to compute the joint solutions at every kth interval and then to interpolate on joint angles (Taylor,1979). The difficulty with this way is that the number k that guarantees a certain deviations, when interpolation is performed, from the desired path is changing from point to point. So a fixed k small enough to ensure a given deviation everywhere in the workspace requires still considerable amount of computation.

Another approach (Benhabib,1986) is to precompute joint solutions for sufficiently large amount of via points in the Cartesian coordinates off-line, and to

interpolate joint angles on-line. This approach may also waste computations and storages, since this sufficient amount is not the same everywhere. In addition, this approach requires us to know the path and corresponding via points well in advance, so that we might execute the motion without much delay.

Then there follows a natural question: What is the number of via points that is minimal yet guarantee a given deviation? An effective solution to this question is found in Taylor's algorithm called *Bounded Deviation Joint Path* (Taylor,1979). This algorithm computes, for a given straight line segments in Cartesian coordinates, nearly minimal number of via points and corresponding joint solutions, in a recursive way.

This algorithm requires one of inverse kinematic methods. Furthermore, it appears to require, if not explicit, a fixed transformation from the Cartesian space to joint space. As we will show in the following chapters, the proposed formula with the kinematic equation has the property of the fixed transformation. Therefore this useful algorithm is expected to help the proposed formula efficiently and accurately control kinematically redundant manipulators. We list the algorithm in Appendix 1. In Chapter 4, we test the algorithm in conjunction with the proposed formula.

On the other hand, there has not been, to our knowledge, any application of the algorithm to the kinematically redundant case. In this sense, the application of the algorithm may be considered an extension of the algorithm to the kinematically redundant case.

2.5 Comparison with Other Methods

In this section, the proposed formula corresponding to equation (2.10) will be compared with two existing null space expressions: the expression used by Luh

and Gu and the extended Jacobian method. The relationships will be investigated on the basis of these comparisons.

2.5.1 Relationship with Luh's Expression

Luh and Gu (Luh,1985) proposed a null space matrix which is defined as

$$\mathbf{N}_r = (\mathbf{n}_1 \cdots \mathbf{n}_r)$$

where $\mathbf{n}_i's$ $(i = 1, 2, \cdots, n-m)$ are orthogonal to every row vector of the Jacobian matrix, so that for each $\mathbf{n}_i$, $\mathbf{J}\mathbf{n}_i = \mathbf{0}$.

Comparing this equation with (2.12), we see immediately that $\mathbf{N}_r$ is a *generic form* of $\mathbf{Z}^T$; or conversely, $\mathbf{Z}^T$ is a specific expression of $\mathbf{N}_r$. Since how to find $\mathbf{N}_r$ was not given, the formulation of $\mathbf{Z}$ in (2.9) provides with a systematic mean to determine $\mathbf{N}_r$.

2.5.2 Relationship with Extended Jacobian Method

Baillieul, in the process of deriving a rate equation called the *extended Jacobian Method*, presented another method to resolve the redundancy, also at the inverse kinematic level (Bailleul,1985,1986). This method derives the additional constraints by using the orthogonality characteristics between the gradient vector of the criteria function and the null space matrix of $\mathbf{J}$ at the optimum, as

$$(\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{h} = \mathbf{0} \tag{2.21}$$

where $\mathbf{J}^+$ is the pseudoinverse of $\mathbf{J}$ as mentioned before.

From the resulting fully specified system of equations, one derives the new Jacobian matrix, called the extended Jacobian, by partially differentiating with respect to the joint variables, just in the same way as in the nonredundant case.

Among the n scalar equations in (2.21), only $n-m$ equations are independent constraints to be determined; for the rank of the null space matrix is $n-m$. The detailed procedures of determining the $n-m$ constraints and their concrete functional forms are not known yet, except for the case $n = m+1$. When $n = m+1$, that is, for the manipulator with just one redundant degree of freedom, a constraint is derived, through a series of procedures, as the following (Baillieul,1985)

$$\begin{aligned} G(\boldsymbol{\theta}) &= \mathbf{n}_J \mathbf{h} \\ &= 0 \end{aligned} \tag{2.22}$$

where

$$\begin{aligned} \mathbf{n}_J &= (\triangle_1, \triangle_2, .., \triangle_n)^T \\ \triangle_i &= (-1)^{i+1} det(J^1, J^2, .., J^{i-1}, J^{i+1}, .., J^n) \end{aligned} \tag{2.23}$$

where $det(.)$ is the determinant, with J^k the k-th column vector of the Jacobian matrix.

It is shown in Appendix 2 that (2.10) reduces to (2.22) in the case that $n = m+1$. Considering this fact and that $\mathbf{Z}$ in (2.10) is a null space matrix of rank $n-m$, we may regard (2.10) as a concrete expression of $n-m$ independent constraints.

2.6 Conclusion

To sum up, we have obtained a closed-form formula for the inverse kinematics of redundant manipulators. It was demonstrated that the proposed formula is concise in form, general in application, and easy to include the desired performance. It was shown that the matrix, $\mathbf{Z}$, in the formula consists of a set of the basis vectors of the null space of the Jacobian matrix. The comparison of the formula to the other methods shows that the formulation provides with a systematic mean to obtain Luh's null space matrix and that the formula serves as a concrete expression for the general case of Baillieul's formulation. Whether the formula is

effective as a kinematic control method will be tested in experiments in Chapter 4.

Chapter 3

Relationship with Resolved Motion Method

3.1 Introduction

In Chapter 2, we proposed a formula that resolves the kinematic redundancy at the inverse kinematic level, by using the null space. The similarity of the formula to the null space term of the Resolved Motion Method led to a conjecture that the former may be the counterpart of the latter at the inverse kinematic level. In this Chapter, we will investigate their relationship and try to prove that the conjecture is correct. Not only for that, we will also derive from the proposed formula the counterpart at the differential level. In addition, we will compare the numerical efficiencies of the two methods, too. In conjunction to the comparison, we will discuss the *repeatability* problem in the Resolved Motion Method, which was pointed out in (Baillieul,1985; Klein,1983), and examine if the proposed formula solves it.

3.2 Resolved Motion Method

In this section, we first introduce the Resolved Motion Method for the redundant case and review its characteristics. Then the causes of repeatability problem in this method will be analyzed.

3.2.1 The Method and Its characteristics

The kinematic equation for manipulators given in (2.1) is again,

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) \tag{3.1}$$

where $\mathbf{x}$ is an m-dimensional vector representing the location of the end effector with respect to the base coordinate system in the workspace, $\boldsymbol{\theta}$ is n-dimensional vector representing joint variables, and $\mathbf{f}$ is a vector function consisting of m scalar functions.

As mentioned in Section 1.1.1, the Resolved Motion Method first derives the differential relationship by differentiating the kinematic equation with respect to time as

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}} \tag{3.2}$$

where the Jacobian matrix $\mathbf{J}$ defined as $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}$ is an $m \times n$ matrix (Whitney,1972). Then by inverting the matrix, we obtain the joint velocity, $\dot{\boldsymbol{\theta}}$. For the nonredundant case($n = m$), $\mathbf{J}$ being square, we can have $\dot{\boldsymbol{\theta}}$ as

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1}\dot{\mathbf{x}}.$$

For the redundant case, however, since $n > m$, $\mathbf{J}^{-1}$ does not exist. Instead, we use a generalized inverse, $\mathbf{J}^{+}$, as (Ben-Israel and Greville, 1980),

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{+}\dot{\mathbf{x}} \tag{3.3}$$

where $\mathbf{J}^+$ is defined as

$$\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}. \tag{3.4}$$

This matrix, known as Moore-Penrose pseudoinverse, has by the way the following properties:

$$\begin{aligned}
\mathbf{J}\mathbf{J}^+\mathbf{J} &= \mathbf{J} \\
\mathbf{J}^+\mathbf{J}\mathbf{J}^+ &= \mathbf{J}^+ \\
(\mathbf{J}^+\mathbf{J})^T &= \mathbf{J}^+\mathbf{J} \\
(\mathbf{J}\mathbf{J}^+)^T &= \mathbf{J}\mathbf{J}^+
\end{aligned}$$

The most general solution to (3.2) is known to be (Ben-Israel,1980)

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^+\dot{\mathbf{x}} + \alpha(\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{h} \tag{3.5}$$

where $(\mathbf{I} - \mathbf{J}^+\mathbf{J})$ is the null space of $\mathbf{J}$, with α a gain constant, and $\mathbf{h}$ an arbitrary vector. Equation (3.5) gives a way to resolve the redundancy at the velocity level. We may call the first term in the r.h.s. a special solution, and the second term a homogeneous solution.

Liégeois(Liégeois,1977) developed a formulation of resolution of redundancy, such that a scalar criteria function may be minimized, by setting to the vector $\mathbf{h}$ the gradient vector of the criteria function, H as in (2.5). He expresses (3.5) in terms of the infinitesimal displacement, as

$$\begin{aligned}
d\boldsymbol{\theta} &= \mathbf{J}^+ d\mathbf{x} + \bar{\alpha}(\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{h} \\
\bar{\alpha} &= \alpha dt
\end{aligned} \tag{3.6}$$

In (3.6)(or (3.5)), at time t, the special solution first moves the end effector to a location $\mathbf{x}(t)$. Then, the homogeneous solution, on the other hand, forces joints to have self-motion to achieve an equilibrium (or optimum), $\boldsymbol{\theta}^*$, where H has a local minimum, for $\mathbf{x}(t)$. In practice, however, it takes a certain amount of time, δt, for the joint variables to reach $\boldsymbol{\theta}^*$, when the end effector has already moved to a new location, $\mathbf{x}(t + \delta t)$ — thus requiring a new $\boldsymbol{\theta}^*$. Therefore, the joint

variables never reach the optimal configuration, but continuously trail behind with a slight difference in the direction of the end effector displacement, $d\mathbf{x}$. In other words, if the end effector would stay at a location sufficiently long, the joints could eventually achieve the optimal configuration corresponding to that location. This analysis appears helpful for understanding the following problem.

3.2.2 Repeatability Problem

The *repeatability* in the robotics context may be defined as the ability to give the same joint values for a given location in workspace, regardless of the path of end effector to that location. In mathematical terms, the repeatability means the *fixed transformation* from the workspace to joint space. The lack of repeatability, of course, can be a considerable drawback in robot manipulators which perform cyclic tasks, because, as the end effector traces the cyclic path, joint variables evolve into states which cannot be predicted in advance.

Klein and Huang (Klein and Huang,1983) noted this problem in (3.3), the Resolved Motion Method without null space term. Baillieul mentioned the same problem in (3.6), the method with null space term (Baillieul,1985). Then what would be exactly the reason for this problem? The above analysis on the characteristics of the method may help understand the reason.

More specifically, the analysis implies that the problem is caused by the following two factors:

1. Because of the *directionality* of $d\mathbf{x}$ in the special solution, the joint variables have different values depending on the *direction* of the repeated path — for instance, a cyclic path — in the workspace. Note, however, that the repeatability can be preserved when tracing only one direction of the cyclic path even in the presence of this factor (Baillieul,1985).

2. Because of the *irreversibility* of the homogeneous solution part, they never return to the initial configuration, once the joint variables reach a steady state trajectory near to the optimal trajectory, $\boldsymbol{\theta}(t)^*$. This situation can happen at the initial transient period when initially guessed joint values are far from the optimum; for the optimal joint values cannot be known in advance.

Beside these factors, as briefly mentioned in Section 1.1.3., another pattern of repeatability problem can exist when, among the multiple joint configurations, transition from one configuration to another happens. About this pattern of repeatability, we will discuss more in detail later.

To sum up, our analysis on the characteristics and the repeatability problem of the method may lead to a conclusion: if we stay in one kind of joint configuration, and if we trace the equilibrium(or optimal) joint values $\boldsymbol{\theta}^*(t)$, then we can preserve the repeatability. Then this conclusion, together with the aforementioned conjecture between the two methods, brings about another question: What is the relationship between the equilibrium states, $\boldsymbol{\theta}^*(t)$, and the states that satisfy Equation (2.11)? The following section gives the answer to this question.

3.3 Relationship between the Two Methods

In this section, we will investigate the relationship between the proposed formula and the Resolved Motion Method. More specifically, we will try to give the answer to the question raised in the previous section: the relationship between the equilibrium states of the Resolved Motion Method and the solutions of Equation (2.11). At the same time, we will derive a rate equation, or differential relationship like the Resolved Motion Method, from Equation (2.11).

3.3.1 Equilibrium State

In the Resolved Motion Method expressed in (3.6), if the end effector stays at a location sufficiently long, the joints arrive at the equilibrium, stopping the self-motion. Hence we have $d\boldsymbol{\theta} = \mathbf{0}$, in addition to $d\mathbf{x} = \mathbf{0}$ — the condition for fixed end effector location. Therefore, from (3.6) we have

$$(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{h} = \mathbf{0}.$$

It is proved in Appendix 3 that this equation holds if and only if (2.10) holds; thus the proposed formula is the necessary and sufficient condition to be satisfied when (3.6) has converged to its equilibrium states. In other words, (2.11) gives the exact equilibrium state, the optimal joint configuration, at which (3.6) will eventually arrive by self-motion for a fixed end effector location. Thus, we may regard (3.6) in the Resolved Motion Method as an approximated equation linearized at states that are exactly determined by (2.11). Note that the above equation is the same as the extra set of constraints proposed by Baillieul (Baillieul,1986).

The practical implication of this relationship would be that we can obtain the equilibrium joint values either by solving (2.11) or by making sufficient iteration of null space term of Resolved Motion Method with fixed end effector location. However, as mentioned in Section 1.1.1, Resolved Motion Method, being a rate equation, is not self-starting: given a location of end effector, corresponding equilibrium joint values cannot be determined without using other methods.

When approaching the equilibrium, the speed of convergence is determined by the value of $\bar{\alpha}$: the larger the value, the faster is the convergence. The value of $\bar{\alpha}$, however, has an upper-limit, above which the equation becomes numerically unstable, not to mention that the manipulator cannot respond because of the torque limitation.

3.3.2 Differential Relationship

The proposed formula and the kinematic equation, in contrast to the Resolved Motion Method, are a set of state equations or equilibrium equations that maps from one space to another. Yet, these equations have in themselves all the necessary informations about *motion* such as velocity and acceleration. Then how do we derive *motion* from these state equations? More specifically, how do we derive the joint velocity from these equations?

This question, obviously, is practically important, because often we need to know joint velocity beside joint displacement. For example, when the dynamic control is needed, or when the motion is specified in terms of end effector velocity, it is more convenient to obtain joint velocity, by using the differential relationship, than to obtain joint displacement.

To derive the joint velocity from (2.11), we have at least two ways:

1. Differentiate the equation with respect to time. Then, the Kinematic equation part becomes the well known Jacobian equation, $\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}}$, while the second part becomes $\mathbf{0} = \mathbf{J}_r\dot{\boldsymbol{\theta}}$, where $\mathbf{J}_r = \frac{\partial \mathbf{Z}\dot{\mathbf{h}}}{\partial \theta}$. Combining these, the resulting equation becomes

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{0} \end{bmatrix} = \mathbf{J}_e\dot{\boldsymbol{\theta}}$$

   where

$$\mathbf{J}_e = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_r \end{bmatrix}$$

   Inverting the above equation gives the joint velocity. Here, $\mathbf{J}_e$ is so called the *extended Jacobian matrix* by Baillieul (Baillieul,1985,1986)

2. Since $\mathbf{J}$ is given, use the pseudoinverse method in (3.6).

The first way, although conceptually simple, turns out to be an inefficient method, because $\mathbf{J}_r$ is a very complicated matrix — see the expression in (Baillieul,1985). In the second way, on the other hand, we need to evaluate $\mathbf{J}^+$ from $\mathbf{J}$, which requires a substantial computation effort — note that we cannot make use of $\mathbf{Z}$ already obtained with another intensive computation.

Instead, we can derive joint velocity by using the following relationship, the proof of which is in Appendix 5:

$$\mathbf{J}^+ = \mathbf{J}_E^{-1} \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \tag{3.7}$$

where $\mathbf{I}_m$ is the identity matrix of rank m and $\mathbf{J}_E$ is defined as

$$\mathbf{J}_E = \begin{bmatrix} \mathbf{J} \\ \mathbf{Z} \end{bmatrix} \tag{3.8}$$

Then the null space matrix is determined as

$$\mathbf{I} - \mathbf{J}^+\mathbf{J} = \mathbf{J}_E^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{Z} \end{bmatrix} \tag{3.9}$$

Substituting these matrices into (3.6), we have

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_E^{-1} \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{Zh} \end{bmatrix} \tag{3.10}$$

By this way, we can make use of the already obtained $\mathbf{Z}$, and thus saving computation time.

3.3.3 Summary

The relationship between (2.11) and (3.6) may be clearly summarized in the following block diagram. The comparison of numerical efficiency for one iteration is treated in the next section, whereas the overall computation efforts are compared in Chapter 4 and Chapter 7.

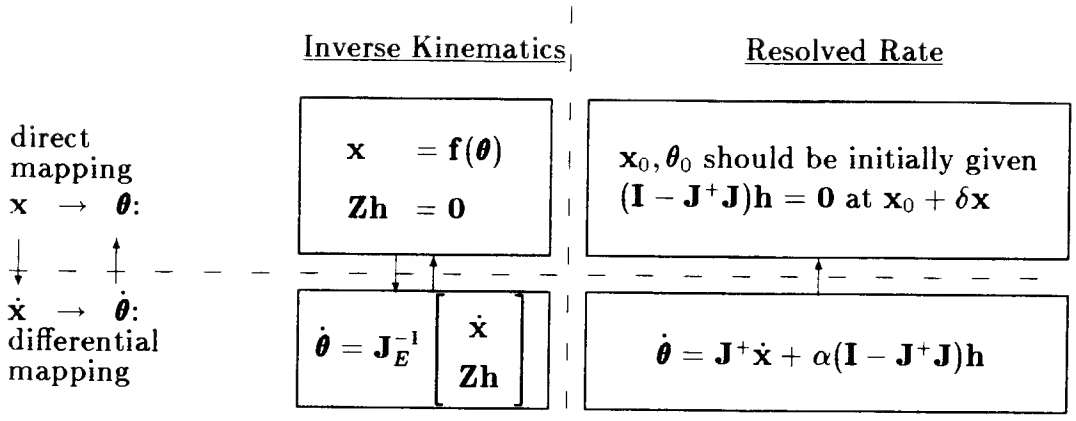


Figure 3.1: The Relationship Between The Proposed Method And The Resolved Motion Method; The Direct Mapping And The Differential Mapping

3.4 Computational Consideration

In light of the relationships between the two methods, each of them can be used as an *alternative* of the other to solve both the equilibrium states and differential variables. More specifically, we can obtain the equilibrium state with the Resolved Motion Method, instead of the proposed method, by iteratively applying the homogeneous part of (3.6). Joint velocities, on the other hand, can be obtained with (3.10) instead of the Resolved Motion Method, too. Then in obtaining these, how do the two methods compare computationally? To answer this, the two methods will be compared in the following subsections.

3.4.1 Obtaining equilibrium States

In comparing the two methods, the following two questions should be answered:

1. *how many arithmetic operations per iteration* does each method require?

2. *how many iterations* are necessary for each method to achieve the same degree of accuracy(and repeatability) from the same initial condition?

The former question will be briefly examined here by first evaluating the number of operations per iteration for the Resolved Motion Method, and then comparing it with that for the proposed method already obtained before. The latter, on the other hand, was partly answered in Section 2.4.4 for the proposed method. Yet since the number of iterations for Resolved Motion Method is not known, the complete answer can be made through a computation example for a special case in the following chapter.

As in Chapter 2, it is assumed that the numerical value **h** is provided at each iteration step with relatively small amount of computation.

The computational effort, N_1, required for the proposed method at each iteration step was obtained in (2.20) as

$$N_1 = \frac{2}{3}m^3 + 11.5n^2 + 2nm - \frac{m^2}{2} - \frac{m}{6} + 96n - 133$$

On the other hand, the total computational effort required for the Resolved Motion Method, N_2, at each iteration step or integration step, may be expressed as

$$N_2 = N_{ev}(N_J + N_a) + N_b \tag{3.11}$$

where N_J is the effort required to compute the Jacobian matrix; N_a, the effort required to evaluate the r.h.s of (3.6), once the Jacobian matrix is given; and N_{ev}, the number of times of evaluations of (3.6); N_b, the effort for numerical integrations. For the general manipulator, N_J is given in (2.18) as

$$N_J = 45n - 93 \tag{3.12}$$

and N_a is evaluated in (Nakamura,1985a) as,

$$N_a = \frac{2}{3}m^3 + nm^2 + 5nm + m^2 - \frac{2}{3}m - n \tag{3.13}$$

N_{ev}, together with N_b, depends on the specific numerical integration method to be used. For example, fourth-order Runge-Kutta method requires that $N_{ev} = 4$ and $N_b = 13n$; while most predictor-corrector methods require that $N_{ev} = 2$ and $N_b = 23n$(Hamming,1973). Considering that N_2 heavily depends on the sum of N_a and N_J, the latter method is much more efficient.

Therefore, N_2 with the predictor-corrector method, is given as

$$N_2 = \frac{4}{3}m^3 + 2nm^2 + 10nm + 2m^2 + 111n - \frac{4}{3}m - 186 \qquad (3.14)$$

On the other hand, Runge-Kutta method requires

$$N_2 = \frac{8}{3}m^3 + 4nm^2 + 20nm + 4m^2 + 189n - \frac{8}{3}m - 372 \qquad (3.15)$$

When $n = 7$ and $m = 6$, $N_2 = 1867$ for the predictor-corrector method and $N_2 = 3503$ for Runge-Kutta method, while $N_1 = 1306$ again. In this comparison, we see that, if the predictor-corrector methods are used for the numerical integration, the Resolved Motion Method requires about 43% more computational effort than the proposed method per iteration, when $n = 7$ and $m = 6$.

3.4.2 Obtaining Joint Velocity

The comparison of computational efficiency of the two differential relationships is quite straightforward: the only differences are the integrands in the r.h.s. of (3.6) and (3.10). Therefore we have only to compare the computational efforts for these terms; the remaining terms are exactly the same.

For the Resolved Motion Method, the effort for the terms, N_a, is again,

$$N_a = \frac{2}{3}m^3 + nm^2 + 5nm + m^2 - \frac{2}{3}m - n$$

Meanwhile, for (3.10), the corresponding part denoted, N_{je}, is obtained as follows:

$$N_{je} = N_{Zh} + N_{Gn} \qquad (3.16)$$

where N_{Zh} is the effort required to compute **Zh**, and N_{Gn} is the effort for Gaussian elimination when inverting $\mathbf{J}_E$. N_{Zh} was evaluated in (2.19), which is again

$$\begin{aligned} N_{Zh} &= N_G + N_s \\ N_G &= \tfrac{2m^3}{3} + \tfrac{3m^2}{2} - \tfrac{7m}{6}; \quad N_s = (n-m)(2m-1) \end{aligned}$$

where N_G is the effort for Gaussian elimination(Nakamura,1985a), and N_s is that for the substitution of Lagrangian multiplier.

Meanwhile, N_{Gn} is obtained simply by substituting m for n in N_G, resulting in

$$N_{Gn} = \frac{2n^3}{3} + \frac{3n^2}{2} - \frac{7n}{6}$$

Hence, N_{je} becomes

$$N_{je} = \frac{2n^3}{3} + \frac{3n^2}{2} - \frac{13n}{6} + \frac{2m^3}{3} - \frac{m^2}{2} - \frac{m}{6} + 2nm \tag{3.17}$$

The total computational effort N_3 for (3.17) was initially given as

$$N_3 = N_{ev}(N_J + N_{je}) + N_b$$

Assuming that the predictor-corrector method is used, and substituting for N_J and N_b respectively, N_3 becomes

$$N_3 = \frac{4n^3}{3} + 3n^2 - 107\frac{5}{3}n + \frac{4m^3}{3} - m^2 - \frac{m}{3} + 4nm - 186 \tag{3.18}$$

Again when $n = 7$ and $m = 6$, $N_3 = 1597$, as compared to $N_2 = 1867$, requiring about 15% less computational effort.

3.5 Conclusion

Summing up, we have introduced the Resolved Motion Method and observed that the deviation from equilibrium states is due to linear approximation characteristics. The repeatability problem was analyzed on the basis of this understanding.

We concluded that the proposed formula provides with equilibrium states that preserve the repeatability. The equilibrium states are also the states at which the Resolved Motion Method is considered as a linearized equation. In addition, the differential relationship was derived from the proposed formula. The computation efforts for one iteration with the two method were compared.

Chapter 4

Numerical Experiment on Kinematic Control Methods

4.1 Introduction

In Chapter 2, we have derived a closed-form formula for inverse kinematics of kinematically redundant manipulators that can achieve additional performances represented in the performance measures. In Chapter 3, we have investigated on the relationship between the formula and the Resolved Motion Method, focusing on the repeatability problem. Although most of the results obtained appear rather obvious, some important points had better be verified through experiments.

In this chapter, therefore, we will make numerical experiments or simulations to confirm some of the important results. To this end, we select a kinematically redundant manipulator and apply (2.10). The resulting system of equations is solved numerically for $\mathbf{x}(t)$, the end effector location. In parallel to using (2.11), the Resolved Motion Method method is applied to the same manipulator with the same tip motion. The points we try to examine or verify through the simulation are as follows:

1. Whether the resulting system in (2.11) gives kinematically correct joint variables for a given $\mathbf{x}$, achieving the additional performance represented by the criteria function we select.

2. How the present method compares to the Resolved Motion Method in terms of accuracy, repeatability, and computational efficiency.

3. Whether the present method gives, in fact, the same equilibrium states as the Resolved Motion Method will eventually reach.

In the comparison of computational efficiency, we consider two situations: when solving the steady state equilibrium states with fixed tip location, and when obtaining joint trajectories with tip moving along the Cartesian path.

4.2 System Description

We use the same manipulator presented in the paper (Yoshikawa,1984) for the sake of comparison with data obtained in that paper: a 3 degrees of freedom manipulator with the end effector moving in the (x, y) plane; thus kinematically redundant. The schematic diagram with necessary parameters is in Figure 4.1.

The desired performance is to avoid singularity. A good criteria function for this objective may be the manipulability(Yoshikawa,1984,1985a), which is given as

$$H = det(\mathbf{J}\mathbf{J}^{\mathbf{T}}) \tag{4.1}$$

The kinematic equation is given as

$$\begin{aligned} x &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\ y &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} \end{aligned} \tag{4.2}$$

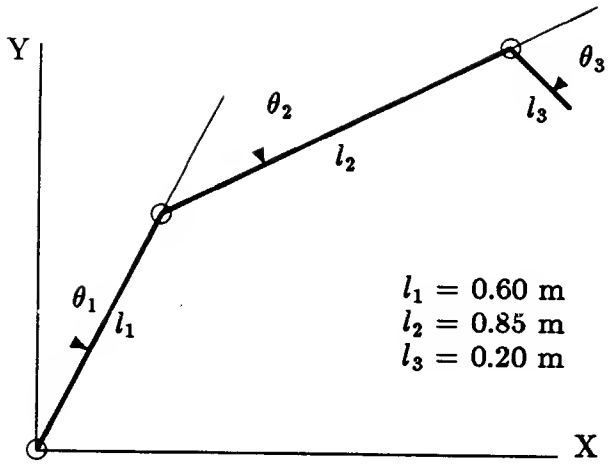


Figure 4.1: The Schematic Diagram Of The Redundant Manipulator

where l_1, l_2, and l_3 represent the length of each link, while the variables with subscripts are denoted in the same way as in Equation 2.13, which is again,

$$s_i = sin(\theta_i), \quad s_{ij...k} = sin(\theta_i + \theta_j + ... + \theta_k),$$
$$c_i = cos(\theta_i), \quad c_{ij...k} = cos(\theta_i + \theta_j + ... + \theta_k), \quad i, j, .., k = 1, 2, 3$$

Then, the Jacobian matrix is obtained as

$$\mathbf{J} = \begin{pmatrix} vc_{123} + uc_{12} + c_1 & vc_{123} + uc_{12} & vc_{123} \\ -vs_{123} - us_{12} - s_1 & -vs_{123} - us_{12} & -vs_{123} \end{pmatrix} \tag{4.3}$$

where

$$u = \frac{l_2}{l_1}, \quad v = \frac{l_3}{l_1}$$

Here, links length are normalized in order to simplify expressions as well as to make the analysis more general.

Equation (4.1), after algebraic manipulations, reduces to

$$H = 2s_3^2u^2v^2 + 2s_{23}s_3uv^2 + 2s_{23}^2v^2 + 2s_2s_{23}uv + s_2^2u^2$$

By partially differentiating, we obtain the gradient vector, $\mathbf{h} = (h_1, h_2, h_3)^T$, with

$$\begin{aligned} h_1 &= 0 \\ h_2 &= 2(c_{23}s_3u + s_{2233})v^2 + 2s_{223}uv + s_{22}u^2 \\ h_3 &= 2(s_{33}u^2 + s_{233}u + s_{2233})v^2 + 2c_{23}s_2uv \end{aligned}$$

Meanwhile, (2.9) is simplified to,

$$\mathbf{Z} = [s_3uv \quad -s_3uv - s_{23}v \quad s_{23}v + s_2u]$$

By applying (2.10), we get

$$\begin{aligned} &2u^2v^3(s_{23}s_{33} - c_{23}s_3^2) + uv^3(2s_{23}s_{233} - 3s_{2233}s_3) + 2u^3v^2s_2s_{33} + \\ &u^2v^2(c_{33} - c_{22}) + 2uv^2(s_2s_{2233} - c_2s_{23}^2) - u^3vs_{22}s_3 - 2u^2vs_2s_3 \\ &= 0 \end{aligned} \tag{4.4}$$

where the same definition as in (4.2) is used for the subscripts.

The system of equations, (4.2), and (4.4) now fully specify the originally underdetermined system of equations, (4.2), while maximizing the criteria H in (4.1). The system of equations may be solved either purely numerically, or by symbolically reducing variables — in this example, θ_1 — and then by using numerical methods. Incidentally, this example suggests that it is possible to reduce variables, thus reducing the order of the system of nonlinear equations, after resolving the redundancy first with all the joint variables.

4.3 Procedures

The task is to trace a square path — thus a cyclic path. As shown in Figure 4.3, the x-y coordinates of the four vertices of the command path in the workspace are successively given counterclockwise from the upper-left vertex as follows:
(446.00, 91.514),(446.00,-8.486), (546.00,-8.486),(546.00, 91.514),
where the units are mm.

The system of equations were numerically solved for joint values, with regard to consecutive equidistant points on the command path — in this example, 100 segments each side — in the counterclockwise direction. As mentioned before, the MINPACK-1 subroutine was used to solve the system of nonlinear equations. The subroutine, by the way, allows only one set of local solutions, among multiple sets of solutions.

Meanwhile, the Resolved Motion Method in (3.5) is also applied to this example for the same path, which the end effector is to track with a constant speed of 10mm/*sec.* (3.5), a system of differential equations, was solved with the fourth order Runge-Kutta integration method, together with the LINPACK subroutines, with the integration time step size, $\Delta t = 0.01$ seconds. We can also use the predictor-corrector method for the better efficiency at the cost of the non self-starting disadvantage and more complex program.

The simulation for the Resolved Motion Method was made with initial joint angle values of (-40.5006, 141.6408, 78.4169) in degrees, which are far from equilibrium states, that were deliberately selected to examine repeatability. The same initial value was also used as the initial guess for the above nonlinear equations for fair comparison of the two methods.

In addition to tracking the square path, another path was traced in order to compare the efficiency in the case when obtaining the joint trajectory is required. The path tried is a straight line path connecting $x = 0.1m, y = 0m$ and $x = 1.6m, y = 0m$, along which the tip reciprocates. The proposed formula is used to obtain joint angles corresponding to the via points determined with Taylor's algorithm mentioned in Section 2.4.4. The via points of the Resolved Motion Method, on the other hand, are simply the equidistant points determined by line segments(the constant tip speed multiplied by the integration time step). Thus, with this method, the equidistant via points are determined either by varying tip

speed or the integration time step. For the proposed formula, hence, the deviation is automatically bounded within a specified value, whereas for the other method the the deviation is controlled by varying the tip speed or time step with trial and error.

4.4 Results

The numerical results of simulations are plotted and listed in Figure 4.2, 4.3 and Tables 4.1, 4.2, 4.3:

- Figure 4.2 is the plot of joint variables solved with the two methods. Note that the 3-D trajectory of joint variables is represented with two 2-D plots: θ_1 vs. θ_2 and θ_1 vs. θ_3.

- On the other hand, Figure 4.3 shows actual trajectories, with the two methods, of the end effector in the workspace, as compared to the command path. The actual trajectory was determined by forward kinematics with joint values obtained by each method.

- Figure 4.4 shows the accuracy and corresponding computational effort to achieve it, with the two methods. The task is to reciprocate a horizontal straight line. Again, the proposed formula was used in conjunction with Taylor's algorithm. Since the Resolved Motion Method may be viewed as a one-time iteration method, the iteration number for this method in the figure is in fact the number of integration steps required to cover a given line-segment.

- Table 4.1 enables one to numerically compare the accuracy of the tip location, and observe the irreversibility factor due to the homogeneous solution term of (3.6), at the transient period. For the purposes, we have selected,

from the data in Figures 4.2, 4.3, five sets of data, which correspond to the consecutive vertices of the square path — counterclockwise from the upper-left one, when the tip begins to trace at the very first cycle.

- Table 4.2 shows the comparison of the accuracy and repeatability when tracing opposite directions in the steady state. In order to obtain the data, the tip was made to reciprocate a straight line segment, which was set to the left vertical side of the square path, after the first two cycles of the square path, when the transient effect appeared negligible.

- Table 4.3 verifies the relationship proved in Appendix 2: the solution with the Resolved Motion Method becomes equal to that with the proposed method after sufficient time, provided that $d\mathbf{x} = \mathbf{0}$. The four vertices were selected for the comparison, as in Table 4.1. To obtain the data with the Resolved Motion Method, the tip was commanded to stop at each vertex, where it made the self motion through iterations of the homogeneous term of (3.6), until the joints converged to an equilibrium configuration. At each vertex, about 25 iterations were needed to converge to the joint values obtained with the proposed method, within the accuracy of 10^{-4} in degree, with $\bar{\alpha} = 10$, which provides about the upper-limit speed of convergence without causing numerical instability.

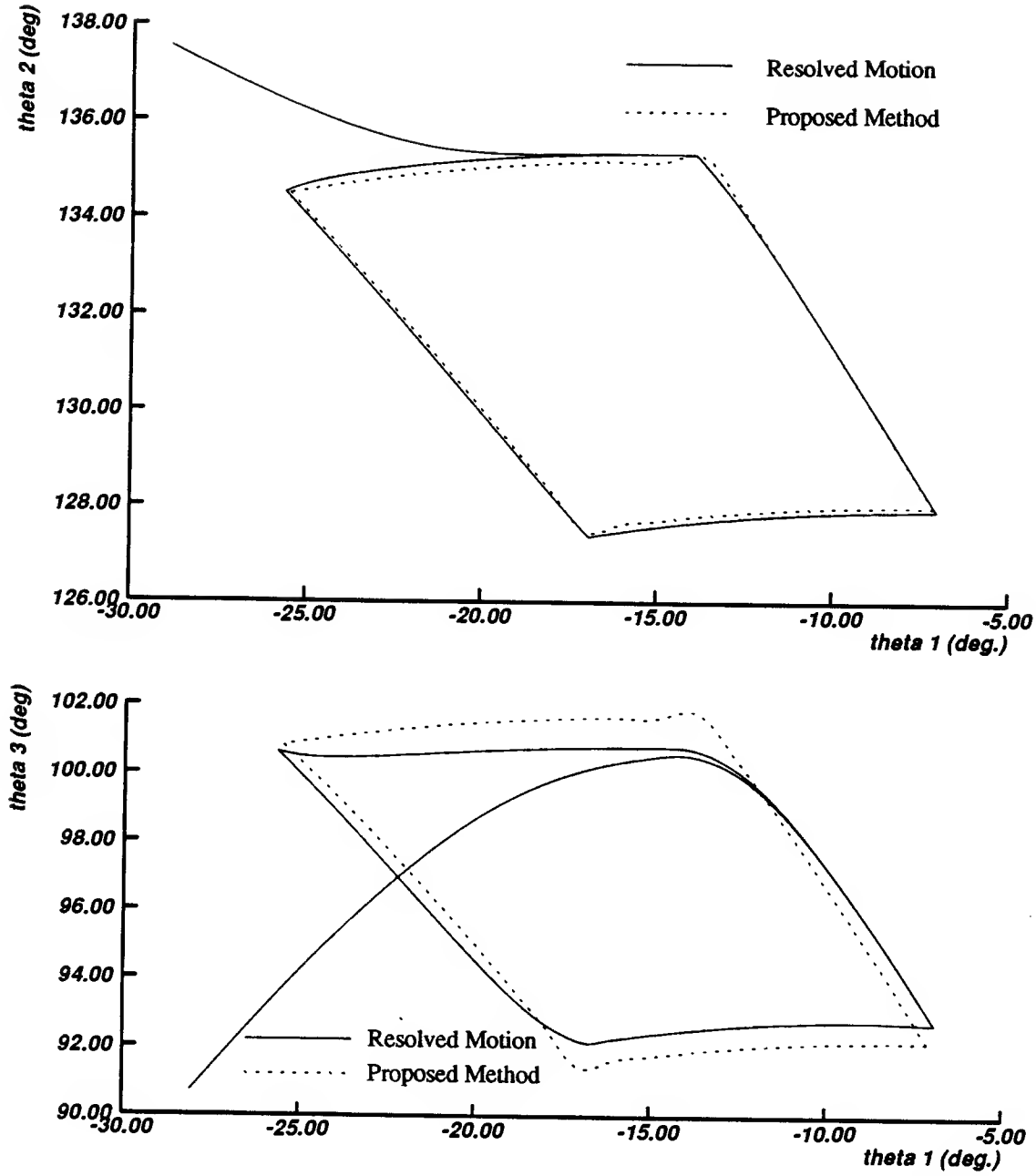


Figure 4.2: Joint trajectories obtained with the two methods: The Three Dimensional Trajectory Is Represented With θ_1 vs. θ_2 and θ_1 vs. θ_3 Trajectories.

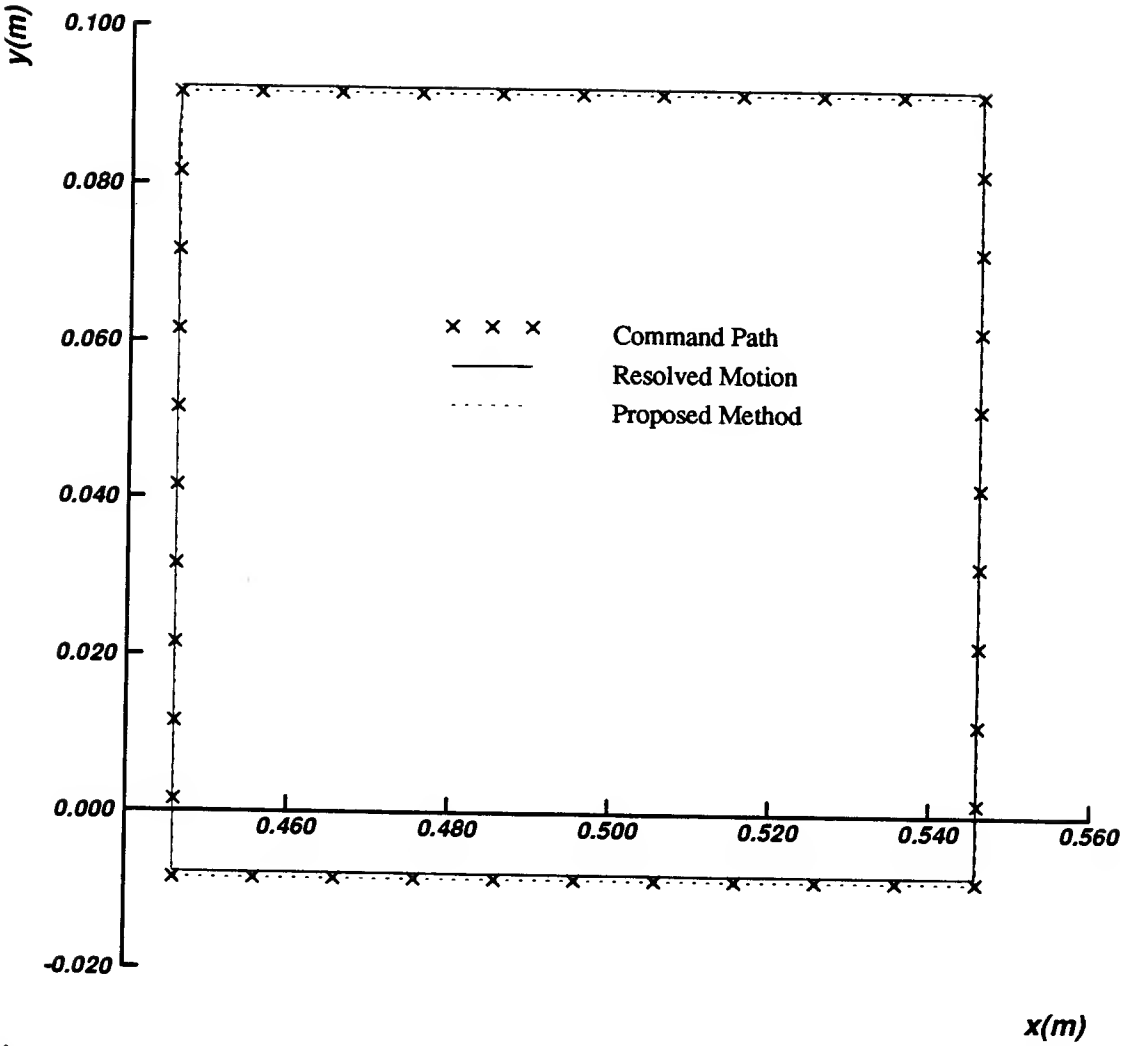


Figure 4.3: The command path and actual workspace trajectories obtained with the two methods

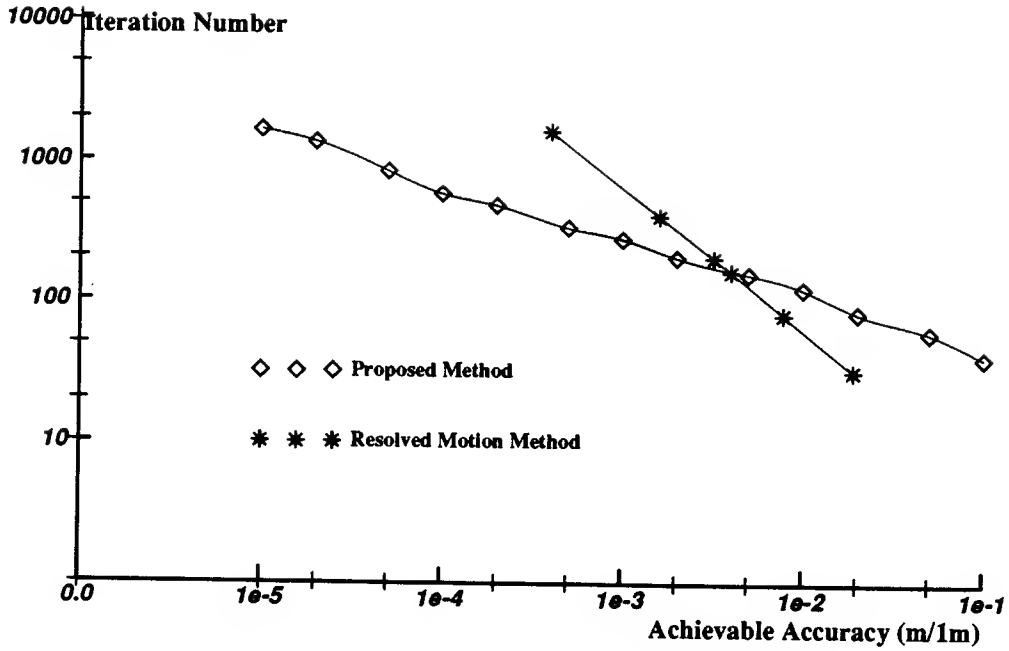


Figure 4.4: The Achievable Accuracy Of Tip-Location With The Two Methods With Relative To The Iteration Numbers.

Table 4.1: The comparison of accuracy and repeatability obtained with the two methods at the first cycle, when transient effect is visible.

	X(mm)	Y(mm)	θ_1(deg)	θ_2(deg)	θ_3(deg)
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
RM Method	: 446.00	91.514	-40.5006	141.6408	78.4169
Command Path	: 446.00	-8.4866			
Proposed Method	: 446.00	-8.4866	-13.4927	135.1801	101.6627
RM Method	: 445.74	-6.5824	-14.0445	135.3160	101.2448
Command Path	: 546.00	-8.4866			
Proposed Method	: 546.00	-8.4863	-7.1232	128.0020	92.1837
RM Method	: 545.52	-6.8216	-7.1924	127.9635	92.4919
Command Path	: 546.00	91.514			
Proposed Method	: 546.00	91.514	-17.0753	127.4846	91.4484
RM Method	: 545.73	92.947	-17.0519	127.3890	91.7938
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
RM Method	: 445.97	93.167	-25.7427	134.4867	100.7127

Table 4.2: The comparison of accuracy and repeatability obtained with the two methods at the steady state, when the tip reciprocates a vertical line-segment.

	X(mm)	Y(mm)	θ_1(deg)	θ_2(deg)	θ_3(deg)
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
RM Method	: 445.89	93.140	-25.7472	134.4929	100.7206
Command Path	: 446.00	-8.4866			
Proposed Method	: 446.00	-8.4866	-13.4927	135.1801	101.6627
RM Method	: 445.63	-6.6244	-14.0476	135.3249	101.2557
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
RM Method	: 445.60	92.912	-25.4178	134.3821	101.2799
Command Path	: 446.00	-8.4866			
Proposed Method	: 446.00	-8.4866	-13.4927	135.1801	101.6627
RM Method	: 445.57	-6.6184	-14.0526	135.3293	101.2610

Table 4.3: The comparison of solutions: Proposed Method vs. RM Method. For RM Method, self-motion was iteratively made by setting dx=0.

	X(mm)	Y(mm)	θ_1(deg)	θ_2(deg)	θ_3(deg)
Command Path	: 446.00	91.514			
Proposed Method	: 446.00	91.514	-25.5116	134.4894	100.8165
RM Method	: 446.00	91.514	-25.5115	134.4894	100.8164
Command Path	: 446.00	-8.4866			
Proposed Method	: 446.00	-8.4866	-13.4927	135.1801	101.6627
RM Method	: 446.00	-8.4868	-13.4927	135.1801	101.6626
Command Path	: 546.00	-8.4866			
Proposed Method	: 546.00	-8.4863	-7.1232	128.0020	92.1837
RM Method	: 546.00	-8.4863	-7.1232	128.0020	92.1837
Command Path	: 546.00	91.514			
Proposed Method	: 546.00	91.514	-17.0753	127.4846	91.4484
RM Method	: 546.00	91.514	-17.0752	127.4846	91.4484

4.5 Discussions

From the results of simulations, we may evaluate the proposed method in terms of accuracy and repeatability by comparing it with the Resolved Motion Method. We can also derive some useful ideas from the relationship between the two methods.

Accuracy

As shown in Table 4.1 and Figure 4.3, the proposed method gives joint variables which exactly correspond to the commanded x and y, while maximizing the criteria function to avoid singularities. Clearly, we see that the accuracy in the workspace achieved with the proposed method is better than that with the Resolved Motion Method.

Therefore, the proposed method provides useful means for accurate position control of the end-effector, when the manipulator is kinematically redundant.

Repeatability

Table 4.2 and Figure 4.2 show that the repeatability is *not* preserved with the Resolved Motion Method because of the two factors mentioned in Section 3.2.2: the initial joint variables which are far from optimal joint values(Figure 4.2) and the dependence on the direction, clockwise or counterclockwise, of the path to be traced (Table 4.2).

On the other hand, it is obvious that the proposed method preserves the repeatability regardless of direction. In other words, the method provides a *fixed transformation* from workspace to joint space. The property of fixed transformation is useful not only for the prediction of joint variables, but also for the precomputation of position dependent terms such as the Jacobian matrix and the

inertia matrix (Raibert and Horn,1978).

Computational Effort

As mentioned before, the comparison of computational efficiency was made in two situations: when solving the steady state equilibrium states with fixed tip location, and when obtaining joint trajectories with tip moving along the Cartesian path. The two cases will be discussed separately.

For Equilibrium State

The number of iterations to achieve the accuracy of 10^{-4} in degree by the proposed method with MINPACK-1 is about from 10. On the other hand, the number of iterations required for the same accuracy by the Resolved Motion Method is about 25 [1] — thus about 2.5 times more iterations than the proposed formula.

It is not immediately clear how the two method will compare for manipulators with more degrees of freedom. Yet, we expect the Resolved Motion Method would require a similarly larger computational effort than the proposed method, considering the following factors:

- As quoted in Section 2.4.3, the proposed formula requires about 15 iterations for manipulators with up to none degrees of freedom. At the same time, it is not quite probable that the Resolved Motion Method requires less iterations for more degrees of freedom case;

- Moreover the former requires about 43% more computation per iteration than the latter.

For Trajectory Generation

[1] This number is based on the Runge-Kutta method. An efficient step-size control with the predictor-corrector method may reduce the number of iterations.

From the simulations, it turned out to be the deviation with the Resolved Motion Method is nearly proportional to the length of equidistant segments. This length in turn is directly proportional to the number of integration steps, N_{step}, or equivalently computational efforts. In other words, for a given straight line segment, $\Delta \mathbf{x}$, the number of integration steps, or iteration number, is estimated as,

$$N_{step} = \frac{\|\Delta \mathbf{x}\|}{\|\dot{\mathbf{x}}\|\delta t}$$

where $\dot{\mathbf{x}}$ is the speed of the tip motion, δt the integration time step, and $\|.\|$ represents the Euclidean norm of vectors. We can reduce, of course, N_{step} by increasing the speed of tip motion and time step at the cost of accuracy and numerical stability.

On the other hand, the deviation with the proposed formula, if Taylor's algorithm is used, decreases exponentially with a base of 2 to 4 as the length between two via points halves — by adding one via point in the middle. Figure 4.5 shows the change of deviation as via points changes.

Thus, as Figure 4.4 shows, with sparse via points, the deviation resulting from the Resolved Motion Method is smaller than that with the proposed method, provided the same computational effort is applied. When the via points become more than a certain number, the situation reverses. In the simulation, the break point is when the number of via points for the proposed method becomes about 17, where the deviation is about $30mm$, assuming total sum of lengths of links is $1m$.

Since the proposed method requires about 9 iterations for a via point, the total number of iterations is about 160. Hence, if the accuracy required is within a deviation of less than $30mm$ per average link length of $1m$, the proposed method gives more accurate trajectory than the Resolved Motion Method.

In the case for more degrees of freedom manipulators, the comparison is ex-

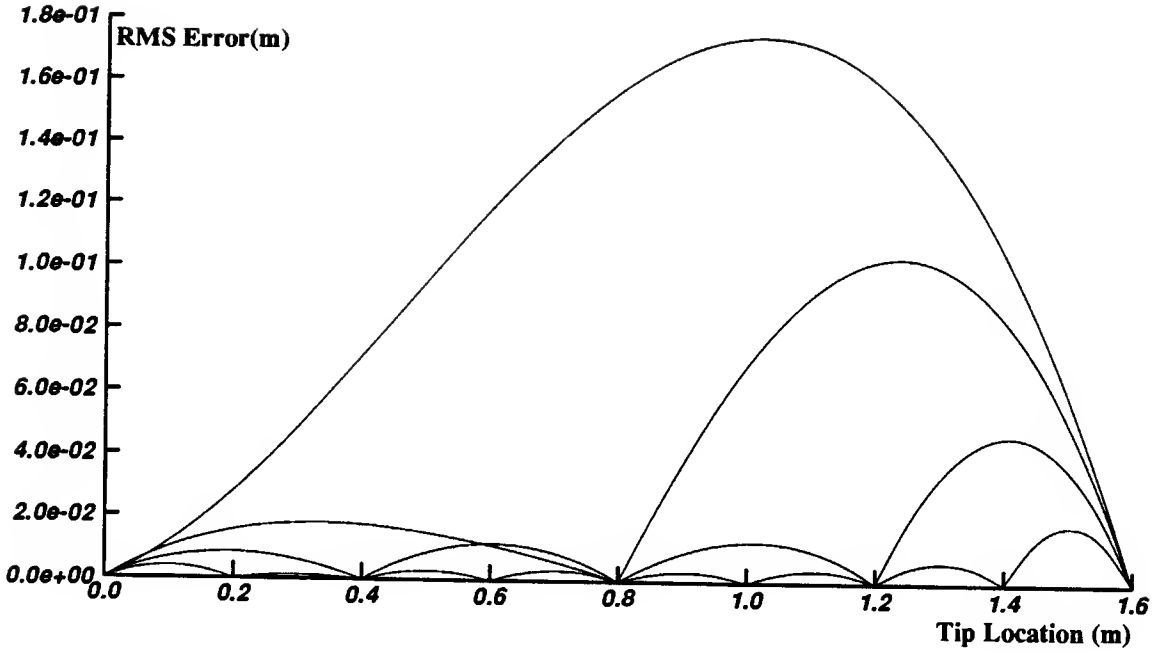


Figure 4.5: The Change of Deviation From the Desired Path, As the Via Points Changes. Note That The Deviation Is Changing Exponentially.

pected to be similar, because of almost the same reasonings made above, when the equilibrium state case was considered.

Relationship between the Two Methods

The result in Table 4.3 shows a nearly perfect agreement of both solutions, verifying that (2.11) in the proposed method is the equilibrium equation at which (3.5) will finally arrive.

Because of the relationship between the two methods, we may use them *interchangeably* as follows:

- The exact equilibrium state can be determined either with the proposed method, or with the Resolved Motion Method by setting $d\mathbf{x} = \mathbf{0}$. The latter, however, would require more computations than the former, as discussed before.

- The incremental displacement $d\boldsymbol{\theta}$, which (3.5) of the Resolved Motion Method easily provides, can be also obtained by using (3.10).

Chapter 5

Development of A Dexterity Measure

5.1 Introduction

The benefits of using a quantitative measure in engineering systems are well known. As mentioned in Section 1.1.3, a quantitative measure provide us with a rational basis upon which we can, without having to rely on experience and intuition alone, analyze, design, and control the systems.

In robot systems, too, various performance measures have been tried, which were listed in Chapter 1. In the subsequent chapters, these measures, in a generic form denoted as H, have been taken into account, when using kinematic control methods for redundant manipulators.

In this chapter and next, we focus on a certain desired performance, *dexterous configuration*, and develop a corresponding measure, *dexterity measure*. This performance, however, becomes in fact quite ambiguous unless the concept of *dexterity* is more precisely defined. The concept of dexterity, when proposed by Klein(1984), appeared to mean (1)the goodness of linear system of the differential

relationship, as indicated by the determinant or condition number of the Jacobian matrix; or (2)a natural appearance resulting from evenly distributed joint angles, represented by summing the squares of the deviation of actuators displacements from their midpoint.

According to this definition, the *least* dexterous configuration would be probably the singular configuration; for at singularity not only is the condition of linear system at its worst, but also awkward appearances happen owing to lining-up or folding of links. In this sense, therefore, dexterity may be viewed as a degree of *farness* or *distance* from singularity. In the thesis, the meaning of dexterity is explicitly specified as the *distance* from singularity.

The distance from singularity, as mentioned in Section 1.1.3, may be represented with one of the following measures derived from the Jacobian matrix: the determinant(or the manipulability measure for the redundant case), the condition number, and the singular values. Yet, in the case of redundant manipulators, it was discussed that these measures cannot explicitly indicate the successive changes in the degree of redundancy. This inability to indicate its change, considering that the degree of redundancy is an important constituent of the distance from singularity, is an obvious shortcoming for distance measures. Furthermore, even for a particular degree of redundancy, there appear relative differences in the distance from singularity that have been unnoticed. Therefore, we feel that a satisfactory dexterity measure should not only include the feature of indicating the change in the degrees of redundancy, but also represent relative differences in a particular degree of redundancy.

In order to better understand the degree of redundancy and the relative differences, we first review in Section 5.2, the concept of singularity, and some basic knowledges about the degree of redundancy. Then, in Section 5.3, we derive a new concept from observation of redundant manipulators. From this concept, a

new performance measure will be developed in Section 5.4. The property of this measure will be discussed in Section 5.5. Finally some concluding remarks will be made in Section 5.6.

5.2 Review of Singularity and Redundancy

This section presents reviews on two basic concepts, singularity and kinematic redundancy, which are necessary for better understanding the distance from singularity in the redundant manipulators.

5.2.1 Singularity

[1] Singularities can be easily observed by examining the differential relationship, (3.2), which is again,

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}} \tag{5.1}$$

where $\boldsymbol{\theta}$ is n-dimensional vector representing joint variables, $\mathbf{x}$ is m-dimensional vector representing end effector location.

For the nonredundant case($n = m$), the Jacobian matrix $\mathbf{J}$ is a square matrix. When the Jacobian matrix becomes singular, a manipulator is said to be at *singular* point.

Hence at singular point, the determinant of $\mathbf{J}$, $det(\mathbf{J})$ equals zero. This simple fact, together with the fact that the determinant is a continuous function of joint variables, provides with some important insights:

1. When $det(\mathbf{J}) = 0$, the rank of $\mathbf{J}$ being reduced, the arm loses corresponding degrees of freedom, with the results that it cannot move in some directions by any combinations of small motions in the joints.

[1] Much of discussions here is based on the personal note on singularity by Professor B.K.P. Horn, the author's supervisor.

2. Thus, as arm approaching this point, small movements in those directions require very large displacements in joint space.

3. At singular points, since the determinant is a continuous function, its sign changes. The change of sign, considering the determinant is the ratio of differential volume of Cartesian coordinates to that of joint coordinates, indicates a change from one kind of solution to another. In fact, just as a change of sign in a continuous function cannot occur without passing through zero, so the arm cannot change from one kind of solution to another without passing through singularity. This property was also discussed in (Uchiyama,1979)

4. At a singularity, thus, two different kinds of solution become one kind; hence, the number of different solutions is reduced.

5. Items 3 and 4 can be explained in terms of Riemann sheets: multiple solutions correspond to multiple sheets each of which represents the mapping from joint angles to Cartesian coordinates; and singular points correspond to the *folds.*

The first two items, by the way, may explain why keeping far from singularity is closely related to dexterity.

From the fact that the determinant becomes zero at singularity, it functions in a sense, as an *indicator* of the presence of a singular point. Besides, the absolute value of the determinant, if geometrically interpreted, represents the *volume* of parallelepiped made of n column vectors (or row vectors) of the Jacobian matrix. This interpretation, together with the fact that at singularity the parallelepiped collapses and the volume becomes zero, is in fact the basis of the idea that the determinant is a *measure of distance* from singularity.

One disadvantage of using determinant, however, as an indicator of singularity is that, once the rank of $\mathbf{J}$ is reduced, it does not distinguish between one state of singularity and another although their remaining ranks are different; the determinants of both of them are equally zero. The more accurate indicator for this purpose would be probably the remaining rank, or degree of freedom, itself.

For redundant manipulators, it was pointed out in Section 1.1.3 that the measure equivalent to $det(\mathbf{J})$ is the manipulability measure, $\sqrt{det(\mathbf{J}\mathbf{J}^T)}$. Yet, this measure, as mentioned in Section 5.1, cannot indicate the change in the degrees of redundancy, which will be reviewed in the following subsection.

5.2.2 Kinematic Redundancy

The degree of redundancy r is formally defined as

$$r = n - m \tag{5.2}$$

where n is the degree of freedom, and m the rank of workspace. In linear algebra (Strang,1980), the degree of redundancy corresponds to the dimension of null space of the Jacobian matrix, the degree of freedom to the dimension of its column space, and the workspace rank to the dimension of its row space. In other words, the degree of redundancy is the maximum number of linearly independent vectors in the null space, $\mathbf{e}_i$, defined as

$$\mathbf{J}\mathbf{e}_i = \mathbf{0} \tag{5.3}$$

But, we find that this definition is not sufficient for describing the concept of *redundancy*.

For instance, consider the following Jacobian matrix representing a three degrees of freedom planar redundant manipulator consisting of three two-dimensional column vectors, J^1, J^2, and J^3 as

$$\mathbf{J} = (J^1 \quad J^2 \quad J^3).$$

If the second and the third links line up, that is $J^3 = cJ^2$ with c any nonzero constant, we know we do not have any redundancy left. However, the null space vector that satisfies (5.3) is obtained as

$$\mathbf{e} = (0 \quad c \quad -1),$$

which is a nonzero vector. Thus, according to the definition, the degree of redundancy is one, whereas the observation indicates there is no redundancy. Then, how can we explain this difference? The discrepancy can be resolved if we modify the meaning of n in (5.2), as the *available* degrees of freedom. But, as mentioned in Section 5.1, it turns out that this modified definition is still inadequate to describe the relative differences in the distance within the same degree of redundancy.

5.3 A New Concept of Distance from Singularity in Redundant Case

In this section, we will observe some Jacobian matrices of kinematically redundant manipulators, and identify relative differences in the distance from singularity. On the basis of the observations, we will propose a new definition of the distance from singularity.

Again, the kinematic equation of a kinematically redundant manipulator is generally given as follows:

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta})$$

where $\mathbf{x} \in \Re^m$, and $\boldsymbol{\theta} \in \Re^n$ with with $m < n$. Then, the Jacobian matrix from the equation, which is given as $\mathbf{J} \in \Re^{m \times n}$, may be denoted in general as,

$$\mathbf{J} = [J^1, J^2, .., J^m, J^{m+1}, .., J^n]$$

where J^k is k-th column vector. If m linearly independent vectors are chosen, without loss of generality, as the first m column vectors of $\mathbf{J}$, then, from linear al-

gebra, the remaining $n-m$ vectors J^{m+1},..,J^n are linear combinations of J^1,...,J^m (Strang,1980).

Observations show that *how many* of these m vectors are included in the linear combination for each of the remaining $n-m$ vectors determines *how far* from singularity a given configuration of the manipulator is at the moment.

To illustrate the point, let us select a manipulator with one degree of redundancy, i.e., $n = m+1$, where

$$\mathbf{J} = [J^1, J^2, ..J^m, J^{m+1}]$$

Consider the following three cases of linear combinations for J^{m+1}:

1. $J^{m+1} = a_1 J^1$

2. $J^{m+1} = a_1 J^1 + a_2 J^2$

3. $J^{m+1} = a_1 J^1 + a_2 J^2 + ... + a_m J^m$

where a_i's, $(i = 1,2,..,m)$ are arbitrary nonzero constants. What are then the differences among these cases?

According to the formal definition in (5.2), the degrees of redundancy for the three cases are the same, one. Alternatively, if the modified definition is used, then the manipulator in Case 1 has no redundancy, whereas those in Cases 2 and 3 have equally one degree of redundancy. However, a careful observation reveals that there still exists another difference in the distance from singularity between Cases 2 and 3. The differences among the three cases may be explained as follows:

1. In Case 1, the manipulator gets into singularity, reducing its rank ($< m$), if *any* two of the first m column vectors happen to line up.

2. In Case 2, singularity arises if *any* two, except for J^1 and J^2, of the m column vectors line up.

3. In Case 3, it preserves its rank ($= m$), although *any* two of the column vectors happen to line up.

In other words, the *chance* for the manipulator to get into singularity decreases by degrees, as the number of linearly independent vectors to be included in the combination increases. These differences in chances of getting into singularity make the *relative differences*, within the same degrees of redundancy, in the distance from singularity. Meanwhile, how many of $J^1, J^2, ..J^m$ appear in each of $J^{m+1}, .., J^n$ uniquely determines the number of distinct combinations of m linearly independent column vectors, or the number of distinct submatrices of rank m in the Jacobian matrix. Hence, this number of submatrices also represents the margin from singularity; the larger the number, the less is the chance of getting into singularity. Of course, the number reduces successively as the number of lining-up of column vectors increases. Of the equivalent two, to determine the number of submatrices would be much easier than to select m vectors and to examine how many of the m vectors are included in the remaining $n - m$ vectors. Note, at the same time, that the observation is not confined to this particular example of one degree of redundancy case, but evidently true for general cases, where the degree of redundancy is more than one.

As another example, consider the following five jointed robot having a three dimensional workspace and thus two degrees of redundancy, where the Jacobian matrix is given as,

$$\mathbf{J} = [J^1 J^2 J^3 J^4 J^5]$$

where J_i's are again the three dimensional column vectors. If J^1, J^2, and J^3 are selected as linearly independent vectors, then J^4 and J^5 are, in general, represented as

$$J^4 = c_1 J^1 + c_2 J^2 + c_3 J^3$$

$$J^5 = d_1 J^1 + d_2 J^2 + d_3 J^3$$

Depending on *how many* and *which* of c_i's and d_i's are zero, we have different numbers and combinations of linearly independent vectors appearing in J^4 and J^5. At the same time, this number and combination of vectors determine the number of submatrices of rank 3 in the Jacobian matrix. The Table 5.1 shows the relationship between the number of submatrices and the number(and the combination) of linearly independent vectors.

It is noteworthy that the number of submatrices *successively* reduces from the maximum, 10, to the minimum, 1, depending on the number and combination of linearly independent vectors. Again, even within the same degree of redundancy, there are different number of submatrices. Clearly this number of submatrices differentiates the relative distance from singularity.

In addition, note that the absolute value of determinant of each submatrix, called *minor*, represents the distance from its own degenerating state. Therefore, the overall distance with relative to singularity should consider the value of each minor of the Jacobian matrix. In other words, in addition to the *number* of submatrices of rank m, the chance of singularity is even less, as *each submatrix* is farther from singularity, that is, as each *minor* has larger absolute value.

The above observations directly lead to a definition of *distance* from singularity as follows:

Definition

The distance from singularity is represented as *the number* of distinct nonsingular submatrices of rank m and the *magnitude* of determinant of each submatrix, or the magnitude of each minor of the Jacobian matrix.

5.4 The Derivation of A New Performance Measure

In this section, on the basis of the distance concept developed, we will derive a performance measure for the kinematic control purpose. More specifically, from the concept the following objectives are to be met to achieve the desired performance:

- to keep *the number* of distinct nonsingular submatrices of rank m as large as possible;
- to make, and at the same time, *the magnitude* of each minor as large as possible.

As an index that explicitly represents these objectives, we propose the following measure:

$$H = \left| \prod_{i}^{p} \Delta_i \right|^{1/p} \tag{5.4}$$

where Δ_i's for $i = 1, 2, .., p$, with $p = n\mathrm{C}m$, are minors of rank m of the Jacobian matrix. Clearly, this measure contains in its expression the two elements of the distance, the *number* and the *magnitude* of distinct minors, in such a way that its increase automatically achieves the above two objectives. To be more specific, since the measure has nonzero values only if all of the minors are nonzero, keeping it greater than zero guarantees the maximum *number* of distinct submatrices. At the same time, since the measure cannot have a large value unless each minor is large, increasing the measure tends to increase the *magnitude* of each, as a whole. Furthermore, since the measure, being a product, becomes smaller if the minors have uneven values — some too large and some too small — it prevents any minor from being dominantly large at the cost of forcing others to be too small.

In (5.4), the exponent $1/p$ is primarily used so that, when $n = m$, the measure might reduce to the absolute value of determinant. We find a similar treatment, in (Yoshikawa,1984), where the manipulability measure is defined by applying the exponent of $1/2$ to $det(\mathbf{J}\mathbf{J}^T)$. Of course, if exponents are used, then physical interpretations for the measure become different. About this, we will discuss more in the following section. Besides, the use of an exponent, when the measure is used in the null space of Resolved Motion Method, results in a different time response of convergence toward the optimal joint configuration. Except for these, the essential characteristics are not changed.

5.5 The Property of The Measure

Examining the new measure, we find the following important properties:

- When $m = n$, i.e., for nonredundant manipulators, the measure reduces to

  $$H = |det(\mathbf{J})|$$

  which is the same as that proposed by (Paul and Stevenson,1983). From a property of the determinant, the measure is conceptually interpreted as the *volume* of a parallelepiped in m-dimensional space, the edges of which come from the rows — or equivalently columns — of the Jacobian matrix, $\mathbf{J}$.

- When $n > m$, thus, the measure represents the *geometric mean* of the volumes of p parallelepipeds made of each combination of m column vectors out of n.

- The points where $\Delta_i = 0$ determine the boundary between one *kind* of joint configuration(or solution) and another kind. These points are also the points where some of the column vectors in Δ_i are linear combinations of the remaining ones in Δ_i, and thereby causing the minor to become zero.

Note that the last property may be considered an extension, to the redundant case, of the nonredundant case in Section 5.2, where the points satisfying $det(\mathbf{J}) = 0$ determines boundaries. This property in fact was used by Borrel and Liegeois(Borrel,1986) to determine the boundaries of different kinds of joint solutions. These boundaries then divide the joint space into subsets, called *aspects*, each of which consists of the kind of joint solution or configuration.

In addition to determining aspects, by using this property, we can *make* the joint configuration stay within a preferred aspect. More specifically, by keeping Δ_i from zero, we preserve the kind of joint solutions for redundant manipulators.

Then why do we need to make the joint configuration stay within an aspect? Some of the reasons have been mentioned in Section 3.2.2, Section 5.2, and Section 1.1.3, which are summarized as the following:

- The switching of aspects can cause a kind of repeatability problem.
- When the switching occurs, discontinuity in motion and awkward configurations may accompany.

Clearly, now that maximizing the performance measure directly prevents Δ_i from zero, it immediately aims at these problems. In other words, by virtue of its property, the new performance measure is expected to help solve these. How the measure achieves the expected performances is one of the main issues in the following chapter.

5.6 Conclusion

To sum up, we have defined the desired performance, the dexterous configuration, and developed the concept of the distance with relative to singularity for redundant manipulators. On the basis of this concept we derived a new performance

measure to help achieve the performance. In addition, from the analysis of the properties of the measure, we expect that the measure may be useful for preventing the repeatability problem and other problems due to switching of aspects.

Table 5.1: The relationship between the number of linearly independent vectors in representing the remaining vectors and the number of submatrices in the Jacobian matrix.

c_1 c_2 c_3	* * *	* * *	0 0 *	0 * *	0 0 *	0 * *
d_1 d_2 d_3	* * *	0 * *	* * *	* * 0	* * 0	0 * *
No. of submatrices	10 ($5C_3$)	9	8	8	7	6
Notes:	A	B	C	D	E	F
c_1 c_2 c_3	0 0 *	0 0 *	0 0 0	0 * 0	0 0 0	0 0 0
d_1 d_2 d_3	0 * *	* 0 0	* * *	0 * 0	0 * 0	0 0 0
No. of submatrices	5	4	4	3	2	1
Notes:	G	H	I	J	K	L

* represent any nonzero value

Notes:

A All of c_i's and d_i's are nonzero.

B Only one among c_i's and d_i's is zero.

C Any two of either c_i's or d_i's zero.

D One of c_i's and one of d_j's are zero with $i \neq j$.

E Two of either c_i's or d_i's are zero and one of the other parts, d_j's or c_j's, is zero with $i \neq j$.

F One of c_i's and one of d_i's are zero with $i = j$.

G Two of either c_i's or d_i's are zero and one of the other parts, d_j's or c_j's, is zero with $i = j$.

H Two of both c_i's and d_j's are zero, with one overlapping $i = j$.

I All of either c_i's or d_j's are zero.

J One of both c_i's and d_j's are nonzero, with $i = j$.

K Only one of either c_i's or d_j's is nonzero.

L All of c_i's and d_j's are zero.

Chapter 6

Relationship and Comparison with other Measures

6.1 Introduction

In this chapter, we test the new performance measure developed in the previous chapter in order to see if it indeed helps achieve the desire performance. Also we will examine if it has the ability of avoiding repeatability problem and problems due to switching of the kinds of solutions.

To this end, first, we investigate the qualitative relationships of the new measure with two major performance measures introduced in Section 1.1.3: the manipulability measure and the condition number. We will also focus on the aforementioned property of preventing the switching of aspects. Then, we will try to confirm, through numerical simulations for three different manipulating situations, the points so far derived.

6.2 Relationship with Other measures

In this section, we investigate the relationship between the proposed measure and each of the two measures: first the manipulability measure and then the condition number.

6.2.1 the relationship to the manipulability measure

From the fact that both represent the distance from singularity, we intuitively see that the new measure and the manipulability measure are somewhat related to each other. However, what is more precise relationship between the two? The following theorem answers the question:

Theorem 3 *For any matrix,* $\mathbf{J} \in \Re^{m \times n}$, *with* $m < n$,

$$det(\mathbf{J}\mathbf{J}^T) = \sum_{i=1}^{p} \Delta_i{}^2$$

where Δ_i*'s,* $i = 1, 2, .., p$, *with* $p = nCm$, *are again minors of rank* m *of the matrix* $\mathbf{J}$.

The proof for the theorem is in Appendix 4. Since the manipulability measure, H_1, is defined as

$$H_1 = \sqrt{det(\mathbf{J}\mathbf{J}^T)}$$

it is expressed in terms of minors as follows:

$$H_1 = \sqrt{\sum_{i=1}^{p} \Delta_i{}^2} \tag{6.1}$$

whereas the new measure expressed in (5.4) is again,

$$H = \left| \prod_{i}^{p} \Delta_i \right|^{1/p}$$

Comparing the two measures, we note the following differences:

1. Geometrically, the manipulability measure may be interpreted as the Euclidean norm of the vector representing the present state in the $\Delta_1 - \Delta_2 - \cdots - \Delta_p$ coordinates system, or as the distance from the origin to the present state in that coordinate system. The new measure, on the other hand, represents the radius of sphere whose volume is equal to that of the hyper-hexahedron made from the Δ_i, $i = 1, 2, \cdots, p$ coordinates in the same coordinate system.

2. As mentioned in Section 5.5, the new measure cannot have a large value if the value of each Δ_i is uneven; whereas the other can have still a large value only if *some* of dominant minors have large values. The manipulability measure can have, in the extreme, some zero minors, as long as the workspace rank is preserved.

   Hence the new measure tends to give more balanced minors than the manipulability measure, not to mention prevents minors from being zero, thus directly controlling over switching of aspects; whereas the manipulability measure does not have an immediate effect on switching of aspects.

3. Note that the manipulability can be also expressed as(Yoshikawa,1985)

$$H_1 = \prod_{k}^{m} \sigma_k$$

   where σ_k is the $k - th$ singular value.

   This expression shows that the measure has exactly the same form as the new measure in that it is a product; the difference is that the manipulability measure is the product of the singular values representing workspace, whereas the new measure the product of minors representing the joint space.

   This difference implies, in a sense, that the former concentrates on preserving the workspace rank while the latter on degrees of freedom of joint

space. Since keeping maximum degrees of freedom of joint space automatically preserves the workspace rank, the latter has the more sufficient yet strict requirements.

To illustrate the second difference, let us consider the same redundant manipulator in Chapter 4, which is to locate the end effector at a certain $x-y$ position. At the end effector location, there exist infinite numbers of configurations or sets of joint values that enable the position, each determining a distinct Jacobian matrix and thus a distinct set of minors.

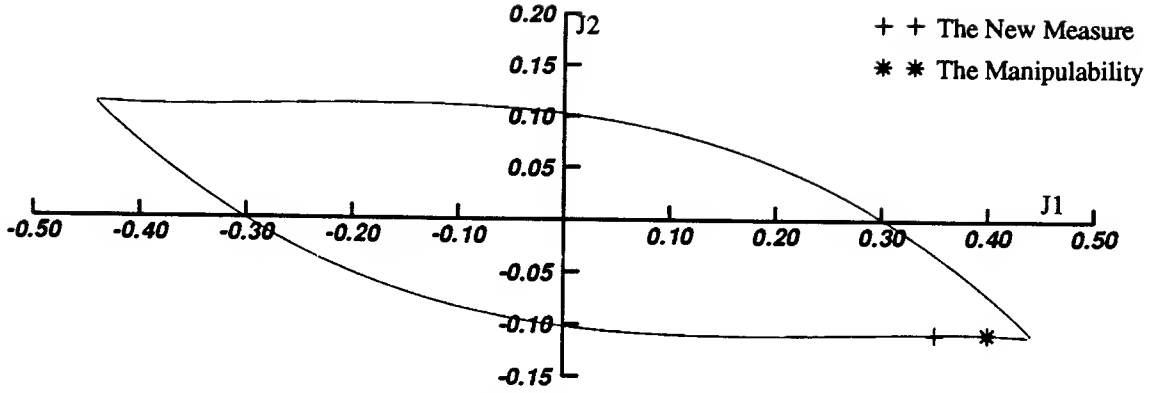
These sets of minors accordingly determine a curve in the $\Delta_1 - \Delta_2 - \Delta_3$ coordinate system, as shown in Figure 6.1. Applying the proposed method for inverse kinematics, when the end effector is located at $x = 0.2m$ $y = 0m$, we can obtain two sets of joint values that maximizes the two measures. Then their corresponding sets of minors are plotted in the same curve in Figure 6.1. From the resulting minor values plotted in the figure, one may confirm the predicted tendency: the new performance measure gives somewhat more balanced minor values than the manipulability measure. This tendency was also confirmed, although different in degrees depending on the locations within the workspace: the more noticeable as the tip moves toward the outer or inner workspace limits.

6.2.2 Relationship with the condition number

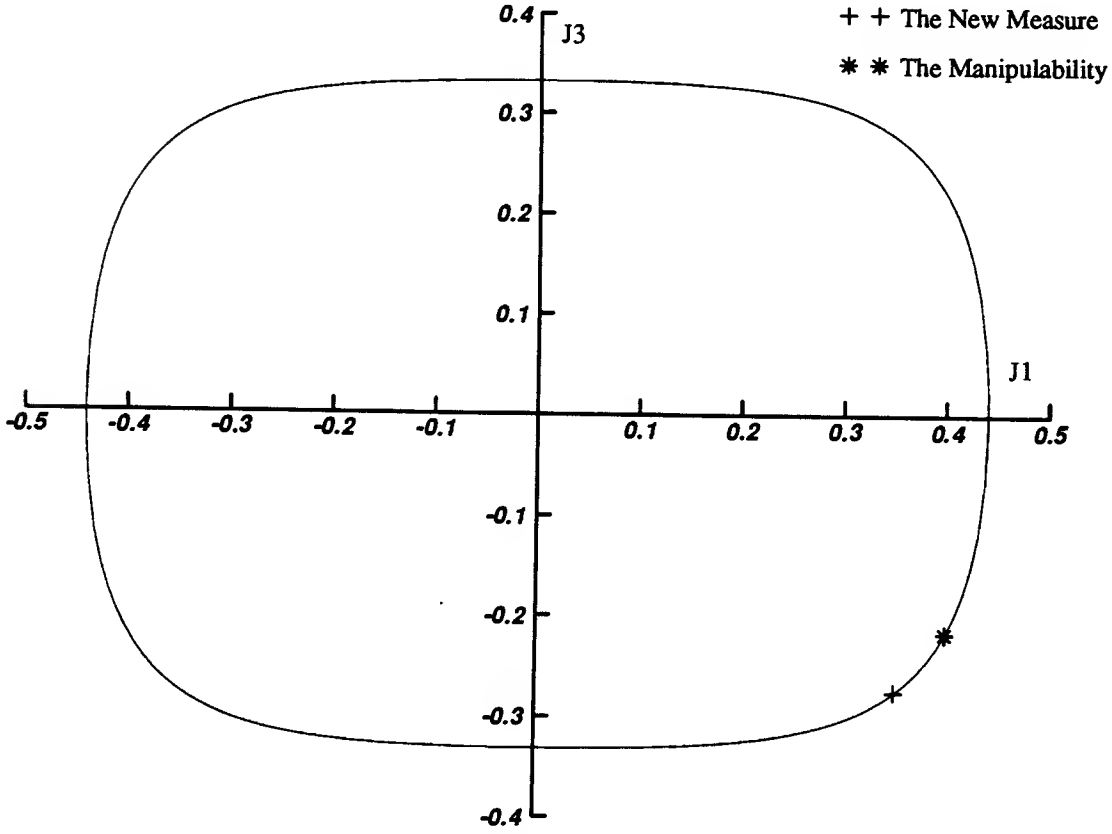
The relationship with the condition number, however, is not so clear as that with the manipulability measure, because of the difficulty in deriving such a pair of simple expressions as (5.4) and (6.1). As well known, the condition number H_2 is defined as

$$H_2 = \frac{\sigma_{max}}{\sigma_{min}} \tag{6.2}$$

where σ_{min} and σ_{max} are minimum and maximum values of singular values, respectively. As pointed out in the above subsection, since the singular values represent



Cross Plot of Minor Trajectory: J1 vs. J2



Cross Plot of Minor Trajectory: J1 vs. J3

Figure 6.1: Trajectories of Minors under the constraint of Kinematic Equation, When the tip is at $x = 0.2, y = 0$; and Two Optimal Configurations with relative to the Two Performance Measure: The Three Dimensional Trajectory Is Represented With Δ_1 vs. Δ_2 and Δ_1 vs. Δ_3 Trajectories.

the workspace rank, this measure, too, concentrates its concern on preserving the workspace rank. More specifically, minimizing the condition number, in effect, results in maximizing σ_{min}. Since nonzero value of σ_{min} guarantees the workspace rank, the measure after all tries to preserve the workspace rank. And as long as it is preserved, it does not concern about what happens within the redundant degrees of freedom.

6.3 Comparison of Performance measures

In this section, the new measure is quantitatively compared with the two other measures. To this end, some numerical experiments are tried for the case of a three degrees of freedom planar manipulator as in Chapter 4.

The points we try to examine through the experiments are as follows:

1. whether the new measure, if used for the kinematic control, helps achieve the desired performance, that is to overcome singularity;

2. whether the measure contributes to preserve *the aspect*, the kind of joint configurations, and how preserving the aspect relates to the repeatability problem;

3. what other effects the transition of aspects brings about.

To examine the first point, the ability to overcome singularity, simulations for the following two cases are tried: when the manipulator has a nearly singular configuration; and when the end effector touches the base. To examine the second and third points, on the other hand, the end effector is made to radially reciprocate between the base and outer limits of the workspace.

For the kinematic control in the experiments, the two equivalent methods, the proposed method or the resolved motion method, were alternately used depending

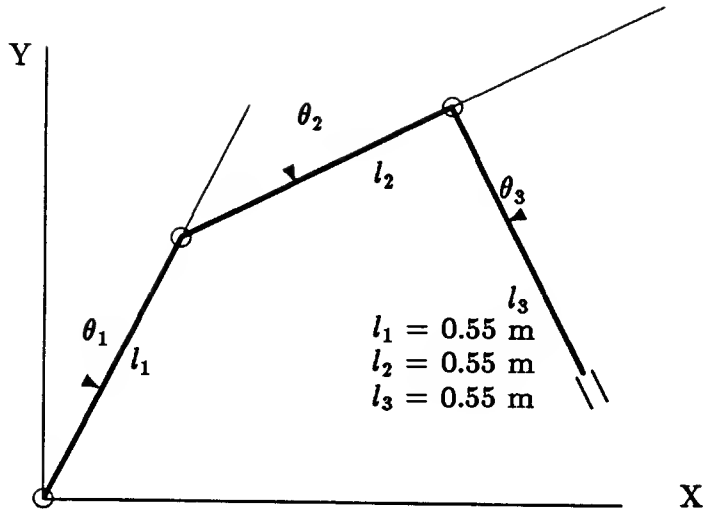


Figure 6.2: The Schematic Diagram Of The Redundant Manipulator With Links of Equal Length

on the necessity and convenience of each experiment.

6.3.1 Overcoming singularity

The two experiments examining the ability of overcoming singularity are made with a manipulator that has three revolute joints with the equal length of $0.55m$, as shown in Figure 6.2. In this case, since $u = 1$ and $v = 1$ in (4.3), the Jacobian matrix becomes

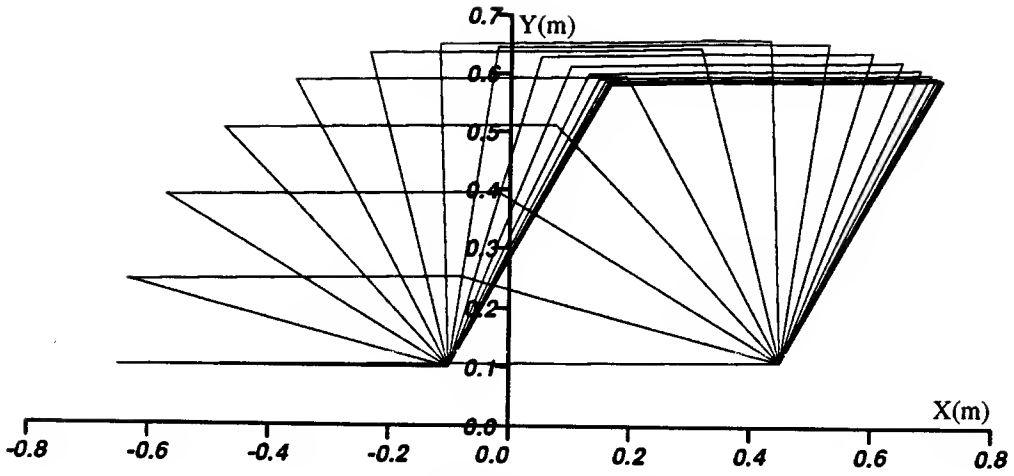
$$\mathbf{J} = \begin{pmatrix} c_{123} + c_{12} + c_1 & c_{123} + c_{12} & c_{123} \\ -s_{123} - s_{12} - s_1 & -s_{123} - s_{12} & -s_{123} \end{pmatrix}$$

Escaping from a nearly singular configuration

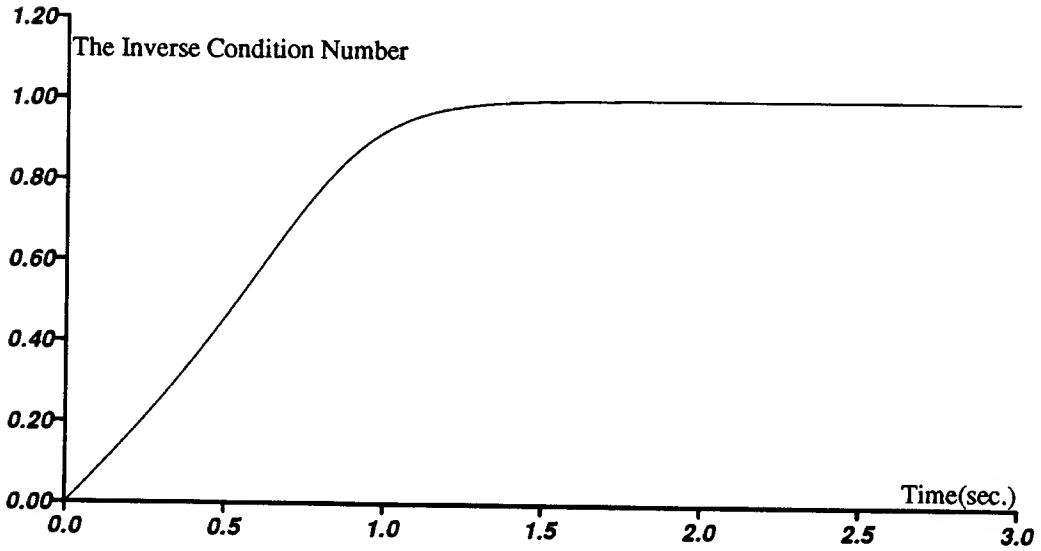
In the first experiment, starting from a nearly singular configuration of $\boldsymbol{\theta} = [-90^o, 179.5^o, 0^o]^T$, the manipulator is made to have the self-motion with each of the three performance measures, and the respective results are compared.

Since the successive change of configurations — as time goes on — is to be examined, the resolved motion method in (3.6), which appears the more suitable for this purpose between the two kinematic control methods, is used with $\alpha = 10$.

Of the measures, by the way, in order to avoid a large value of null space term, the condition number is not directly minimized; instead, its inverse is maximized. That is because the condition number has a large value at the initial state, resulting in a large value of $\mathbf{h}$ — it is proportional to the condition number — which, in turn, causes a very large value of null space term.

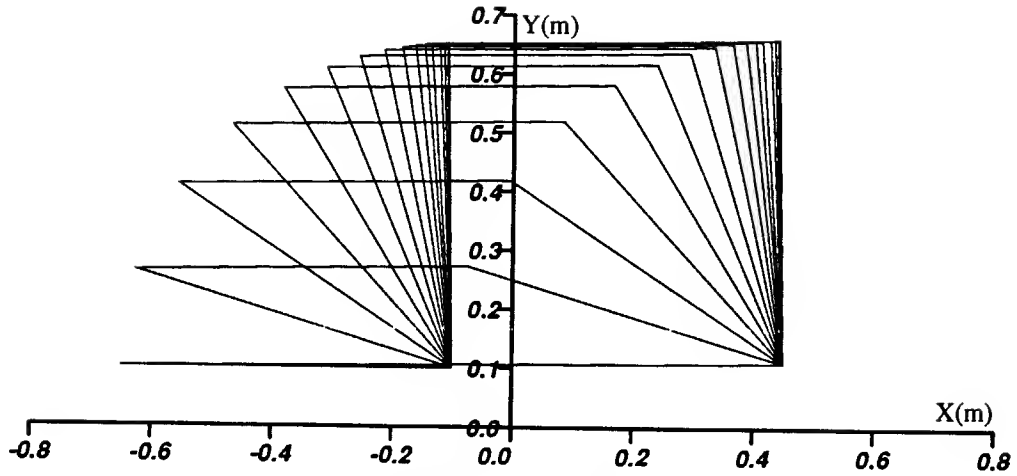


The Condition Number Is Used

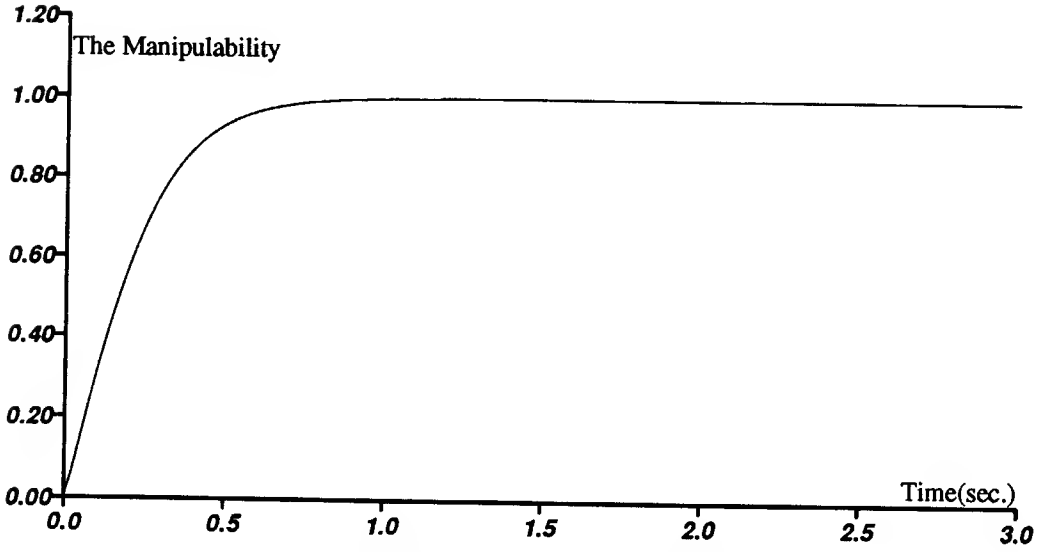


Time Response of The Inverse Condition Number

Figure 6.3: The Singularity Avoidance Ability When The Condition Number Is Used: The Configuration Is Changing Toward The Optimal Configuration As Time Goes On. The Link Lengths Are $l_1 = l_2 = l_3 = 0.55m$.



The Manipulability is Used



Time Response of The Manipulability

Figure 6.4: The Singularity Avoidance Ability When The Manipulability Measure Is Used: The Configuration Is Changing Toward The Optimal Configuration As Time Goes On. The Link Lengths Are $l_1 = l_2 = l_3 = 0.55m$.

The result of the experiment is given in Figures 6.3, 6.4, 6.5, where the change of configurations and the time response of each measure are shown. From this result, it is clearly demonstrated that each measure, if included in the null space term, makes the manipulator get out of the singular configuration, driving joint values toward the state where the measure has the maximum value for the tip location. The effect of including the performance measures becomes even more obvious, if the self-motion *without* the measures is considered, when no self-motion happens at all because the r.h.s of (3.6) is zero. The speed of convergence for the condition number case is noticeably slower than those of the two other measures, both of which are almost the same.

At the steady state, whereas the condition number has a considerably different one, the new measure and the manipulability measure have optimal configurations that look surprisingly similar. Yet a close inspection shows that they are slightly different. The reason for these similar configurations is as follows. The optimal condition, $\mathbf{Zh} = \mathbf{0}$, for each measure is in general quite different from one another. Even for the manipulator in Figure 6.2 with such a particularly symmetric geometry, joint solutions for each measure are different. However, the above optimal condition for each measure turns out to be satisfied only at locations (x, y) of end effector satisfying $x^2 + y^2 = l^2$, with a particular joint values of $\theta_2 = \theta_3 = 90^o$ or $\theta_2 = \theta_3 = -90^o$. Here l is the value of equal length of the three links of given manipulator.

When passing the base

Whereas the previous experiment examines the self-escaping ability, the present one examines the behavior of the manipulator when the tip touches the base, forming a closed kinematic chain — a triangle. This case is of interest because intuitively we see the self-motion is not possible, except for rigid body rotation

of the triangle with respect to the base, as long as the tip stays at that location. This intuition can be easily confirmed if the projection matrix, $\mathbf{I} - \mathbf{J}^{+}\mathbf{J}$, in the homogeneous solution term of (3.6), is symbolically derived. To be more specific, when the tip is located at the base, the Jacobian matrix in (3.2) is modified as the following symbolic form:

$$\mathbf{J} = \begin{pmatrix} 0 & j_{12} & j_{13} \\ 0 & j22 & j23 \end{pmatrix} \tag{6.3}$$

from which the projection matrix can be derived as,

$$\mathbf{I} - \mathbf{J}^{+}\mathbf{J} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{6.4}$$

Clearly, through the projection matrix, only θ_1 is affected by the gradient of performance measures, $\mathbf{h}$, resulting in a rigid body rotation of the triangle with respect to the base, without making any other changes in the configuration.

Then, is it impossible for the manipulator to get into and out of the configuration? In other words, can we obtain the inverse kinematic solution that resolves the motion when the tip is passing the base? The answer for the question is that, although the homogeneous term becomes ineffective with the tip at the base, it is still possible for the manipulator to get into and out of the point. The reason for the answer may be analyzed as follows:

- When the tip *approaching* the base, the homogeneous solution term, although diminishing, still exists, continuing the effort to achieve the optimal configuration, until the very point of the base.

- When the tip *getting out* of the base, on the other hand, now that the projection matrix is given as in (6.4), the homogeneous term does not contribute to overcoming the closed chain configuration. Yet, since the rank is

still preserved, the pseudoinverse $\mathbf{J}^+$ is available, which can be derived from (3.4) as

$$\mathbf{J}^+ = \begin{pmatrix} 0 & 0 \\ j_{23} & -j_{13} \\ -j_{22} & j_{12} \end{pmatrix} \tag{6.5}$$

Since, in this matrix, the second and the third row vectors are linearly independent, we may have a differential tip displacement in *any* direction we like in the workspace.

- Then, once the tip is apart from the base — no matter how small the distance may be — the homogeneous term immediately begins to restore its effectiveness.

To sum up, at the base, where the particular solution term is still well defined, this term drives the tip out of it, while the homogeneous term is momentarily ineffective; at the remaining region in the workspace, on the other hand, both terms are effective. Furthermore, between the two regions, the respective transitions of the two terms are smooth without discontinuity.

One may suspects that the overcoming ability could have come from the *inexact* tip location — the tip can be slightly off the base — due to the linearization characteristics of (3.6). On the other hand, inexact Jacobian matrix may have made it possible for the tip to get out of the base, which, with the exact Jacobian matrix, might be impossible. But these are not the cases, because both the Jacobian matrix in (6.3) and the pseudoinverse in (6.5) are exact expressions defined at an exact point, the base. Rather, the ability comes from intrinsic back-up function of kinematic redundancy.

The aforementioned analysis is well confirmed in the following experiment. In the experiment, the tip is made to move along the straight line starting from $\mathbf{x} = [0.2, 0]^T$ to $[-0.2, 0]^T$, passing the base, $[0, 0]^T$, with units in meters. Together

with the tip motion, the three measures are included in the inverse kinematic method proposed in Chapter 2, which was proved to give the exact equilibrium solution, thus excluding the effect of inaccuracy in both the tip location and the Jacobian matrix.

From the result shown in Figures 6.6, we see that with the three measures the manipulator had no difficulty to treat the point. We may conclude that the use of redundant manipulator seals the hole in the workspace at the origin, where, without the kinematic redundancy, singularity is unavoidable.

In the Figure, one may note the smoothness of motion when the new measure is used, as compared to motions with the other measures: When the other measures are used, the motions approaching the base from $\mathbf{x} = [0.2, 0]^T$ are rather jerky — see the abrupt changes in θ_1. The reason for the smoothness is not very clear right now; but we can guess that keeping minors to be balanced prevents the abrupt changes in the joint angles.

6.3.2 Preserving the aspect and its effect to the repeatability problem

In the following experiment, we examine whether the manipulator, with the new performance measure, can preserve the aspect indeed, by comparing to the cases with other measures. The manipulator to be used for the purpose is the same one as used in the previous chapter: the three degrees of freedom planar manipulator with revolute joints of $l_1 = 0.6, l_2 = 0.85$, and $l_3 = 0.2$ with units in meters.

In the experiment, the tip is made to reciprocate radially between the base and the outer limit, where the manipulator fully extends. Here, the tip motion is so made, not because the radial motion itself is of primary concern, but because it is a way of scanning the workspace to examine the ability to preserve the aspect. To be more specific, a series of configurations corresponding to the tip reciprocating

in one radial direction represent, owing to rotational symmetry, those in all the other directions, thus covering the whole workspace. Of course, the rotational symmetry lies in the fact that the new performance measure, together with the other measures, depends on θ_2 and θ_3 only, and is independent of θ_1 — hence, one optimal configuration for a fixed tip location is symmetrical to another for the other tip location of the same distance from the base.

Of the two kinematic control methods, the proposed method, being the more convenient one for obtaining the equilibrium states, is mainly used together with the three measures. When applying the method, to solve the successive joint configurations as the tip reciprocates, the present joint values are used as the initial conditions for the next tip location. The very first joint configurations corresponding to the starting point of the tip, by the way, are determined by obtaining the global minimum. To do this, we first find out, for comparison, all the local minima by providing with every possible initial condition for solving the system of nonlinear equations, (2.11). These initial conditions, in turn, may be determined by finely tessellating the joint space, that is, the domains of joint variables, $\theta_1, \theta_2, \theta_3$. In parallel to this, all the local maxima at each tip location are obtained with the proposed method, in order to cross-examine if the successive generations of joint values are indeed correct. In addition to the joint configurations, corresponding minor values are obtained to examine the correlation between joint configurations and minor values.

In Figures 6.7, 6.9, 6.11, the optimal joint configurations based on each of the performance measures and the value of each measure are plotted.

Corresponding minor values are plotted in Figures 6.8, 6.10, 6.12.

As shown in the figures, each performance measure has two distinct *sets* of configurations that, depending on the tip location, alternately give the global optimum. Hence, each of the two sets of configurations has its own corresponding

performance measure curve: the one, consisting of mostly the ladder-shaped configurations, corresponds to the curve with a solid line; and the other, consisting of mostly 'N'-shaped configurations, to the curve with a broken line. For convenience, let us denote the former configurations A, and the latter configurations B.

Besides, note that the mirror-image sets of Configurations A and B with respect to the x-axis, although having the same values of the corresponding measure, are not included in the figures, for brevity. Of course, there are still additional sets of configurations that, corresponding to local maxima instead of global maxima, are not shown in the figures either.

What do we find from the resulting configurations? Let us examine the configurations generated by each performance measure one after another.

The Manipulability Measure

Figure 6.7 shows the two configurations and corresponding values of performance measure, when the manipulability measure is used. In the figure, depending on the tip location, each of the two configurations alternately assumes larger performance measure value than the other. In the region between $x = 1.1m$ and $x = 1.6m$, however, the two configurations become identical, having the same performance measure values. Then what happens with the two configurations, when the tip is coming out of this region of the identical configuration? To answer this question, we need more careful observations as follows.

In Configurations A, the initial configuration is preserved within almost the whole workspace except for the region between the base and $x = 0.1m$. That is, except for this region, the initial shape does not change, regardless of wherever the tip location may be, and to whatever direction — toward or outward from the base — the tip may move.

In Configuration B, on the other hand, where the tip starts from near the base, the initial configuration is preserved only if the tip is located within a certain distance from the base, about 1 m; outside the location, the configurations *shift* or *merge* into Configuration A. And once merged, configurations corresponding to subsequent tip motions stay within Configuration A, never returning to Configuration B.

Here, we observe that the manipulator has switched the aspect or the kind of joint solutions. Moreover the configuration, once switched from one aspect to another, does not return to the initial configuration: the expected repeatability problem. Then, what happened to the minors when this switching of aspect occurred? Did they change their sign indeed, passing through zero values? Figure 6.8 clearly shows that is the case: the tip location where merging happens is the place where one of the minors changes its sign.

The Condition Number

In the case of the condition number, the situation is even more complicated. In this case, Configurations A consists of successive joint configurations, where the tip starts from near outer limit of the workspace toward the base, whereas Configuration B represents the movements in the opposite direction from the base. Differently from the manipulability case, in *both* Configurations A and B, the initial configurations are merged into the other. Furthermore, the locations where mergings occur are different: about $x = 0.7m$ for Configuration A; and $x = 1.3m$ for Configuration B. And within $x = 0.7m$ or outer than $x = 1.3m$, the two configurations are identical, giving the same measure curve.

Hence, the repeatability problem occurs, between $x = 0.7m$ and $x = 1.3m$, when the tip reverses its direction after experiencing a merging. Again, the minor curve in Figure 6.10 shows that, when the switchings occur, signs of minors

change.

The New Measure

When the new performance measure is used, there still exist two distinct configurations. One thing particularly noticeable is that there is no switching for both configurations; there is no repeatability problem at all. Because of no merging effects, the initial configurations are distinctly preserved showing also distinct measure curves. As expected, the minor curve in Figure 6.12 clearly shows that there is no sign change at all for the three minors. We see from this an obvious consequence of using a measure that has a direct control over each minor value; the property mentioned in Section 5.5 has an immediate effect.

Besides, we can observe, as compared to cases with the other measures, smoother movements near base, which remind us of the previous experiment results when passing the base.

6.3.3 Discontinuity effects

As mentioned in Section 5.5, when merging of configurations or switching of aspects occur, discontinuous joint motion was predicted.

To confirm the prediction, we obtained joint velocities of both configurations for each measure. Here the tip is made to move with a constant velocity of $0.1(m/sec)$. To resolve the velocity, we used the Resolved Motion Method.

Figure 6.13, 6.14, 6.15 show the resulting velocity curve corresponding to the configurations obtained in the previous subsection. As expected, the two configurations of the new measure and Configuration A of the manipulability measure, where there is no switching, show quite smooth joint velocity trajectories as plotted in Figures 6.13 and 6.15.

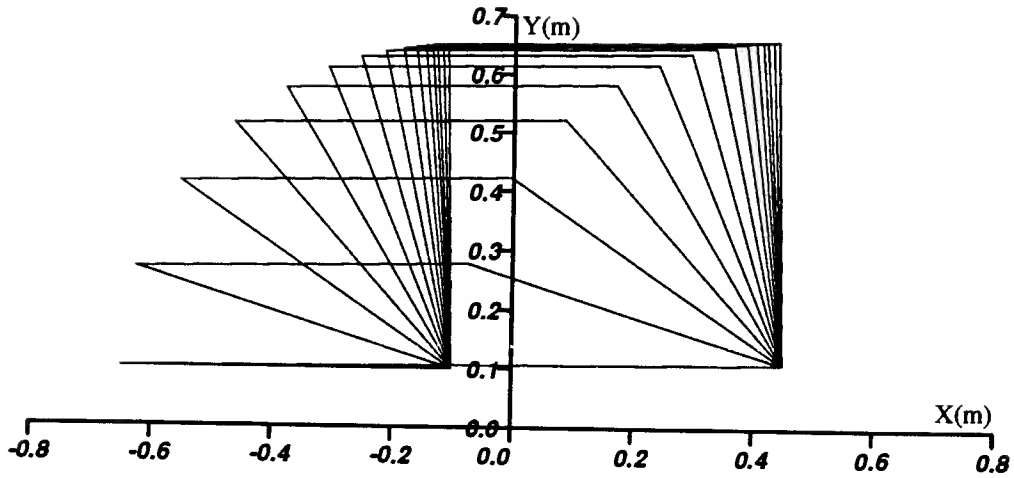
For the remaining cases, where switchings occur, the velocity trajectory is generally rugged, confirming our prediction. Yet, the degrees of ruggedness for the two measures are different. A careful observation reveals the following: for the manipulability measure, the merging happens through some intermediate configurations, reducing the degree of discontinuity; whereas, for the condition number, the switching happens instantaneously with few intermediate states, resulting in much larger values of joint velocity.

It is not very clear right now why this difference exists. We will probably be able to get some clue, if the gradients of the two measure are first expressed in symbolic forms and then each term is examined to pinpoint the cause of the abrupt switching. To have the symbolic expressions for this inspection appears to be possible, although quite complicated, in this particular manipulator case.

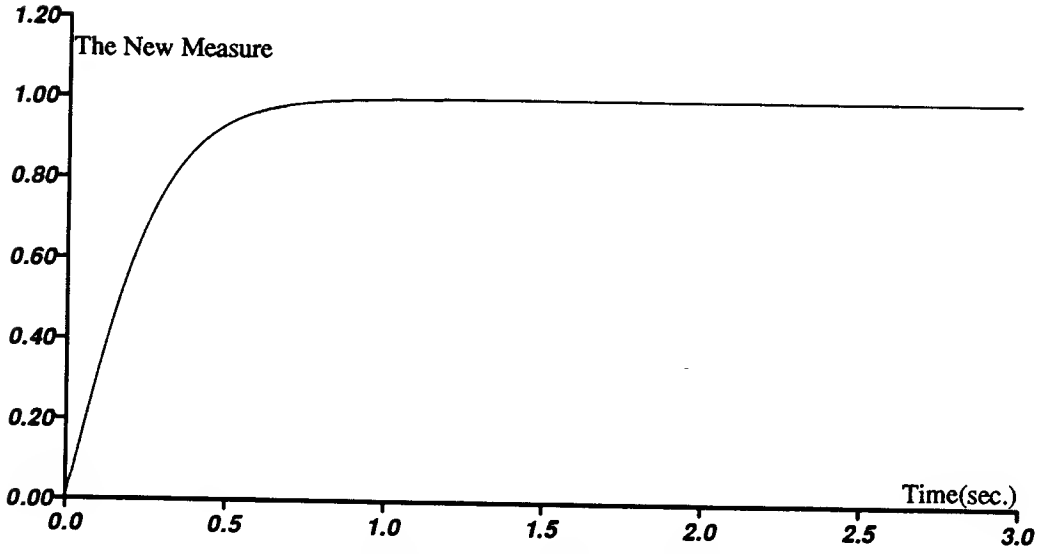
6.4 Conclusion

Summing up, the new measure has been compared with the two other measures, both qualitatively and quantitatively. The result of analysis on qualitative relationship agrees well with the experimental results.

To summarize these results, all the measures showed the ability to overcome singularity by successfully treating the two locations: the location corresponding to almost straight line configuration and the base or the origin. The essential difference between the new measure and the other ones is its ability to explicitly prevent the minors from becoming zero. This ability in effect prevents the merging of configurations and switching of aspects, which in turn enable to prevent the repeatability problem and impulsive motions. In addition, the extra care, built in the new measure, about the balanced values of minors appears to contribute to noticeably smooth movements near the base.

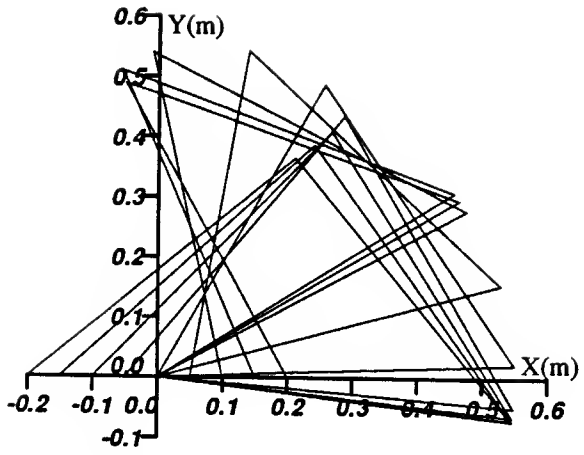


The New Measure Is Used

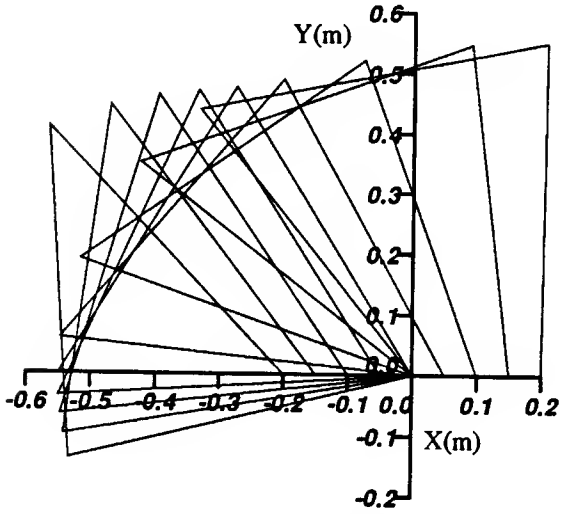


Time Response of The New Measure

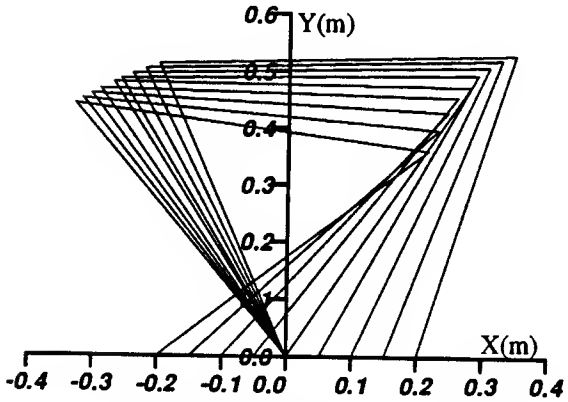
Figure 6.5: The Singularity Avoidance Ability When The New Measure Is Used: The Configuration Is Changing Toward The Optimal Configuration As Time Goes On. The Link Lengths Are $l_1 = l_2 = l_3 = 0.55m$.



The Condition Number Is Used

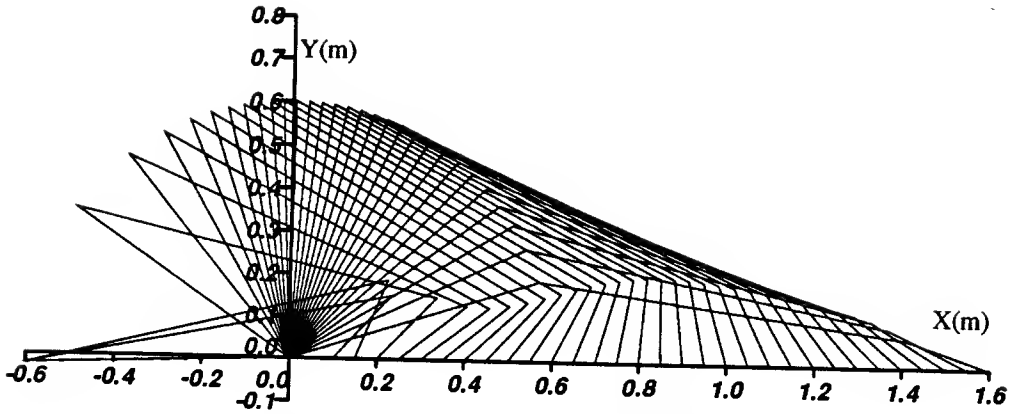


The Manipulability is Used

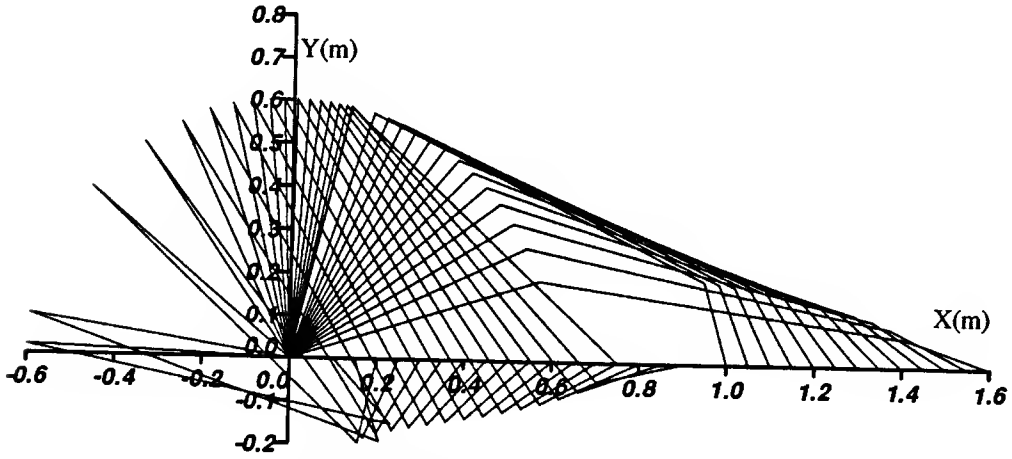


The New Measure Is Used

Figure 6.6: Use Of The Three Performance Measures, When The Tip is Passing The Base From $x = 0.2$ to $x = -0.2$ On The x-axis, With Link Lengths, $l_1 = l_2 = l_3 = 0.55m$.



The Manipulability Is Used In Configuration A



The manipulability Is Used In Configuration B

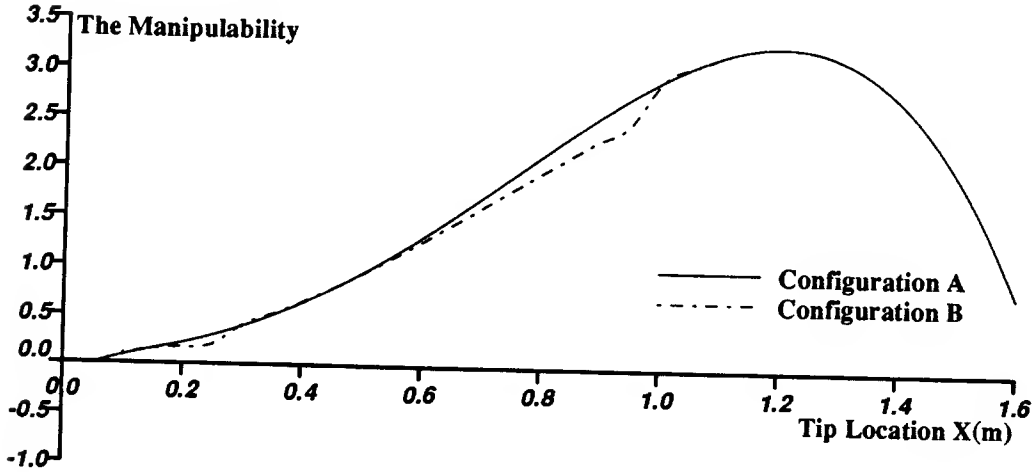
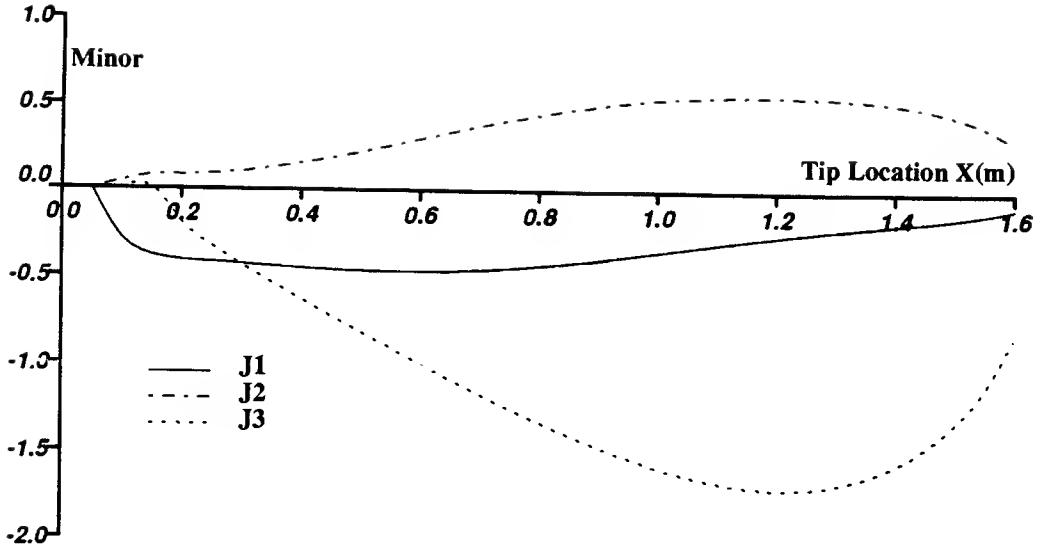
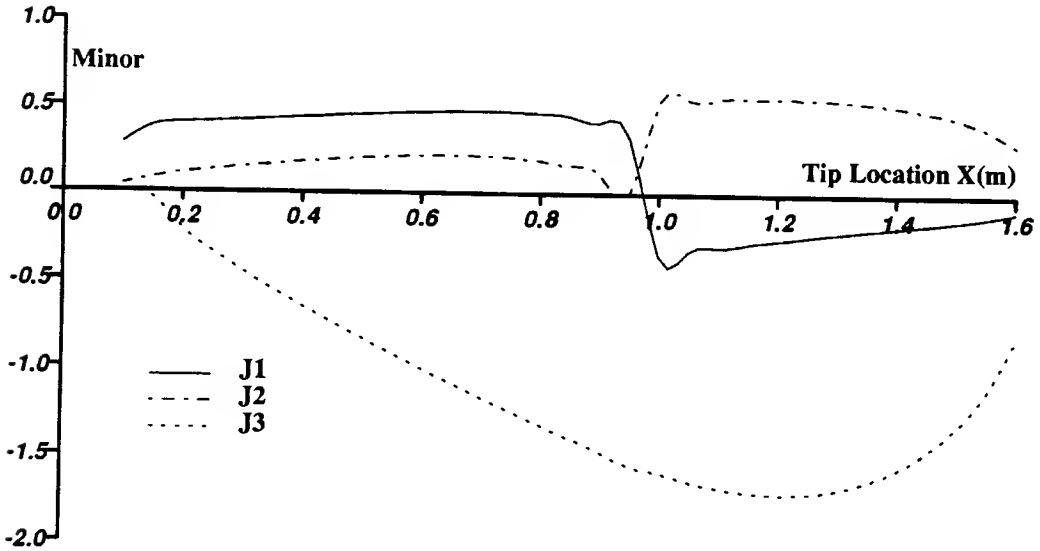


Figure 6.7: Configurations A And B, When The Manipulability is Used As Performance Measure, And Corresponding Values of The Measure, Where A And B Refer To The Tip-Motion With Different Initial Configurations.

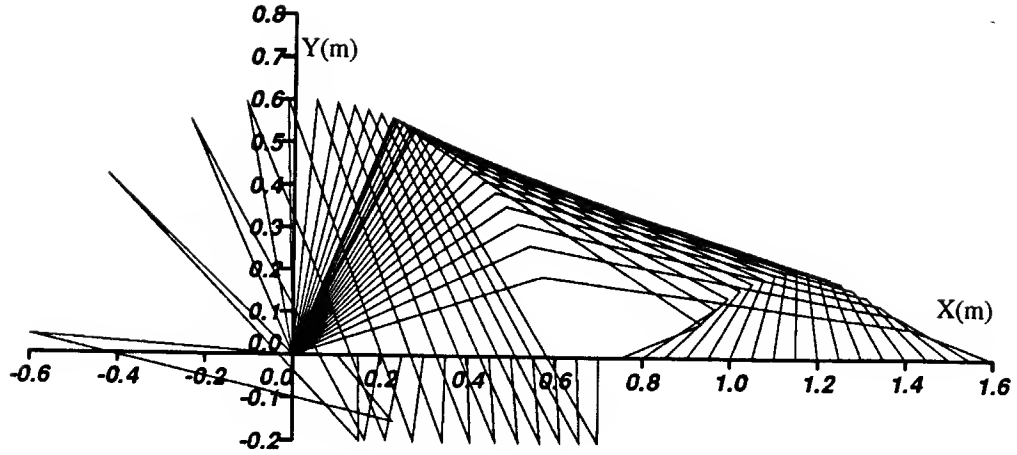


Minors For Configuration A

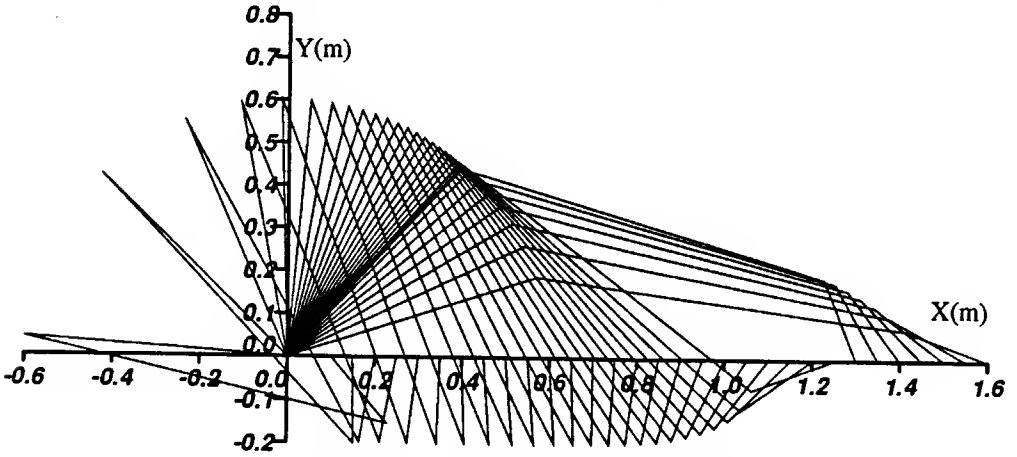


Minors For Configuration B

Figure 6.8: The Minor Values In Configurations A And B, When The Manipulability is Used As Performance Measure, Where A And B Refer To The Tip-Motion With Different Initial Configuration.



The Condition Number Is Used In Configuration A



The Condition Number Is Used In Configuration B

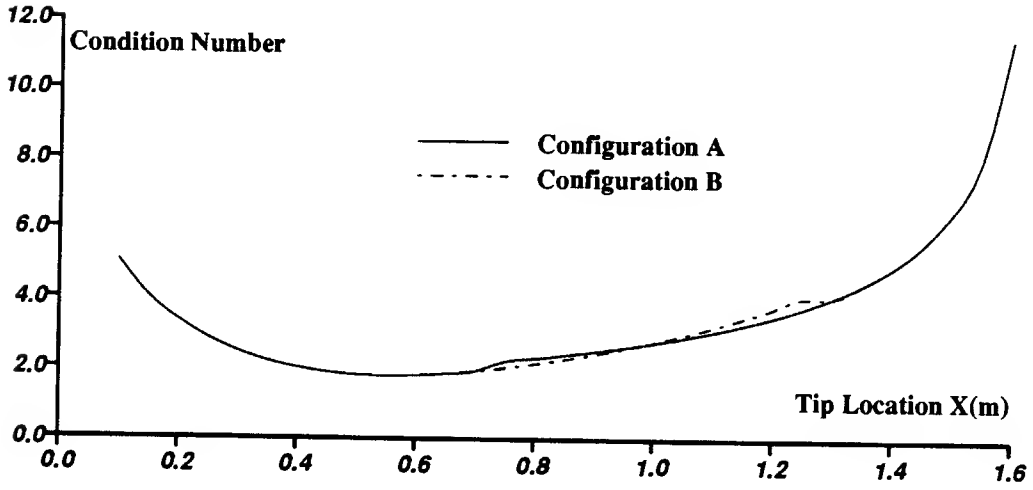
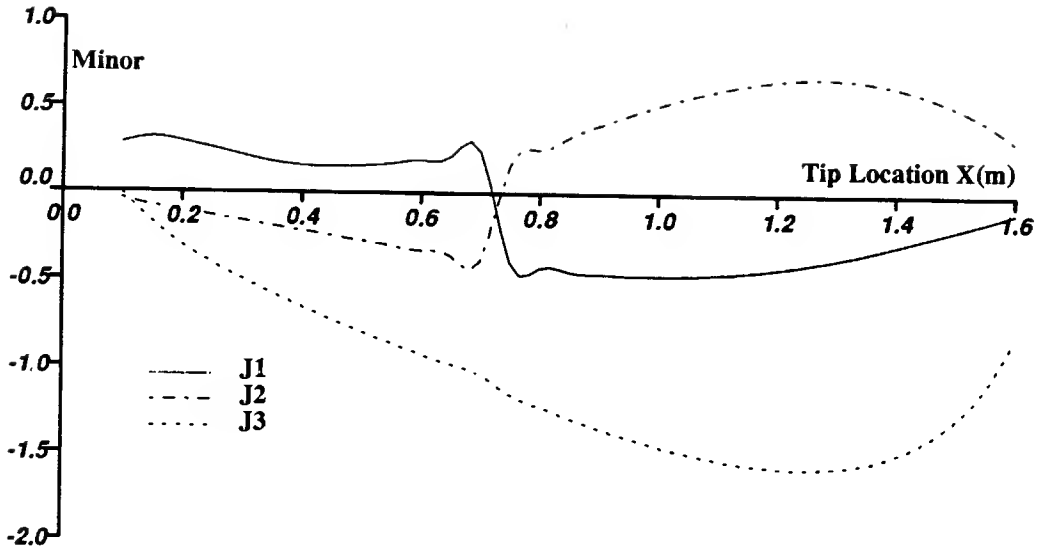
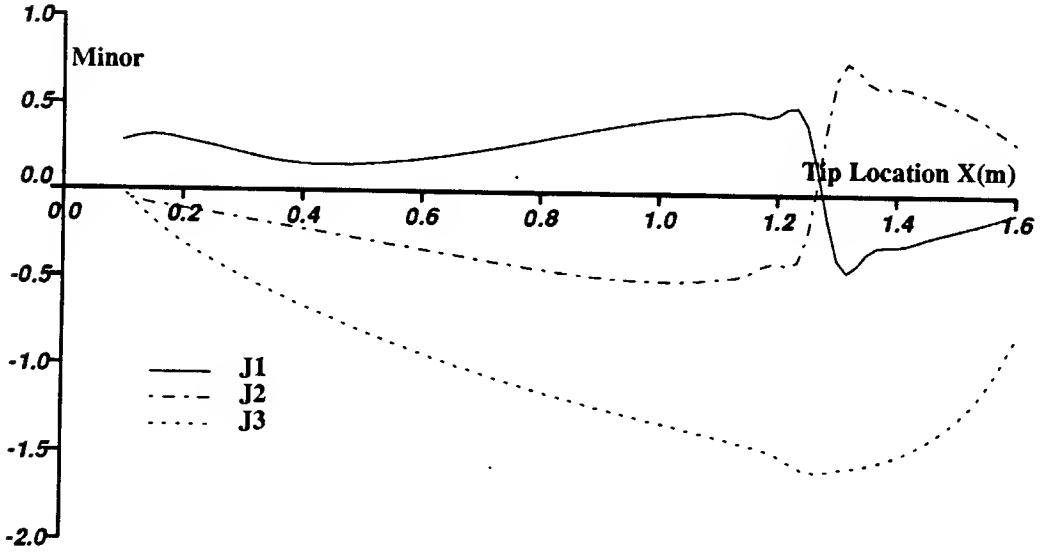


Figure 6.9: Configurations A And B, When The Condition Number Is Used As Performance Measure, And Corresponding Values of The Measure, Where A Refers To The Tip-Motion From 1.6 to 0.1 And B From 0.1 to 1.6.

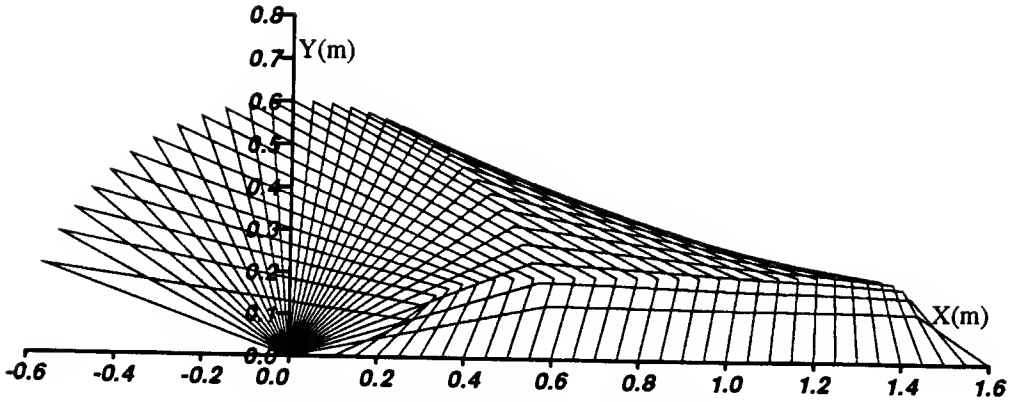


Minors For Configuration A

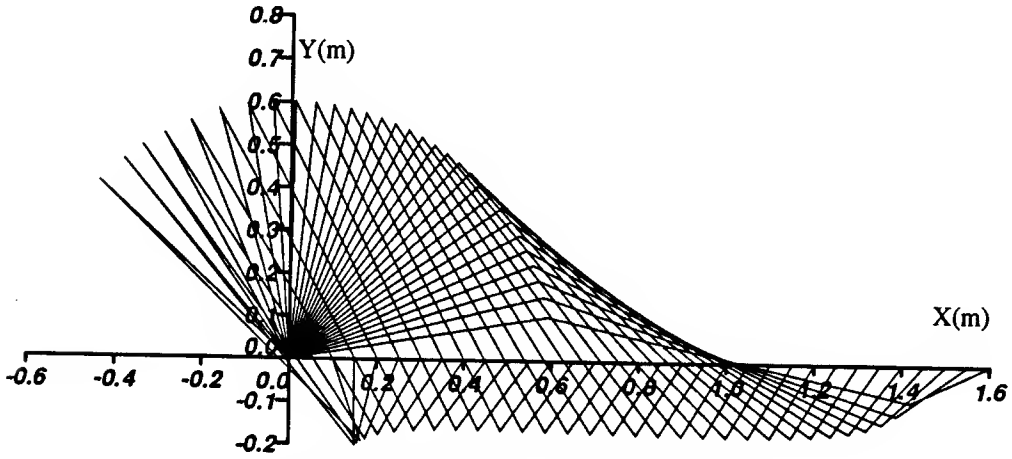


Minors For Configuration B

Figure 6.10: The Minor Values In Configurations A And B, When The Condition Number is Used As Performance Measure, Where A And B Refer To The Tip-Motion With Different Initial Tip Location.



The New Measure Is Used In Configuration A



The New Measure Is Used In Configuration B

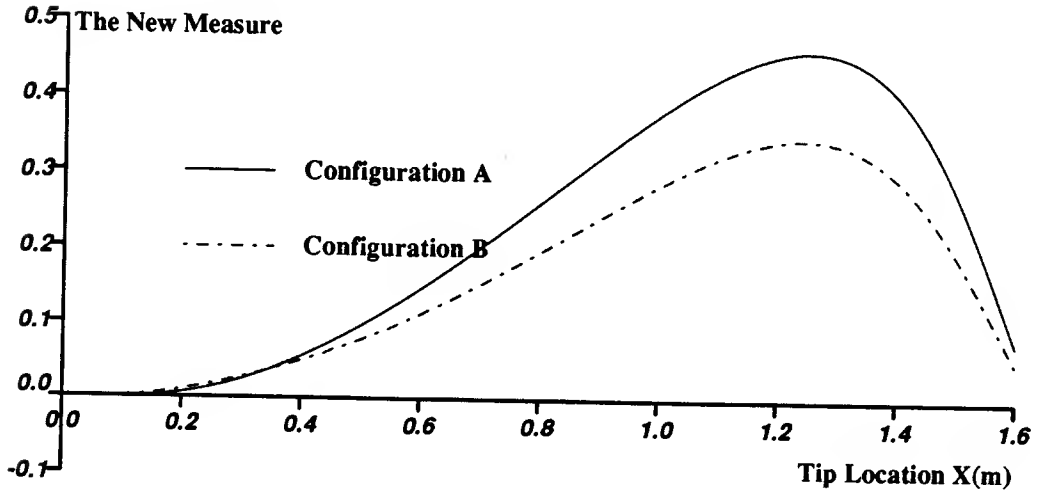
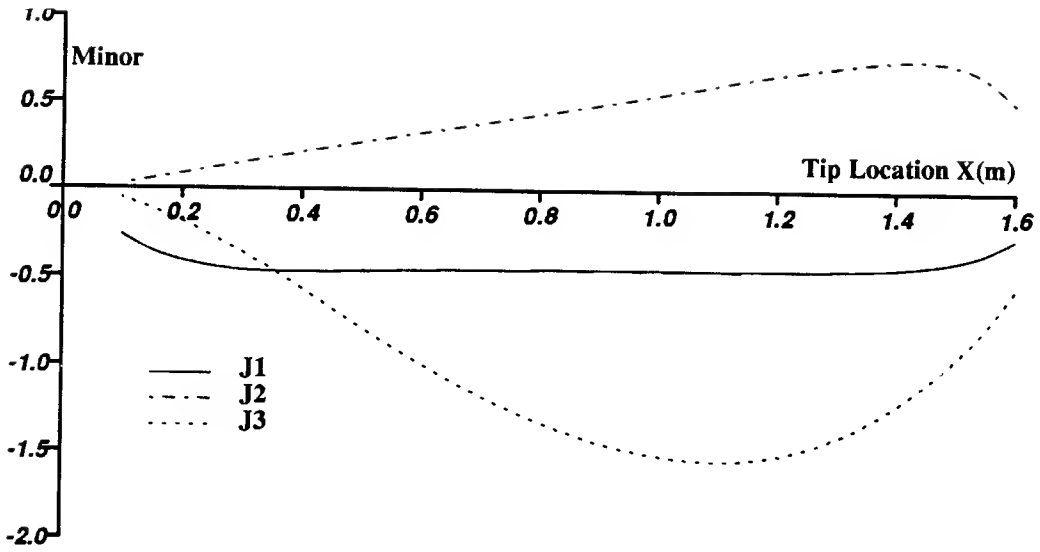
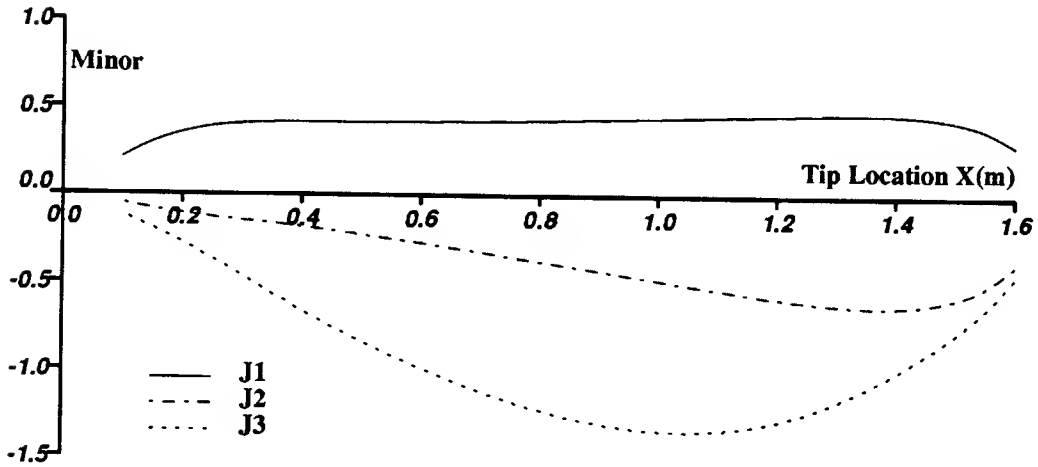


Figure 6.11: Configurations A And B, When The New Measure Is Used As Performance Measure, And Corresponding Values of The Measure, Where A And B Refer To The Tip-Motions With Different Initial Configurations.

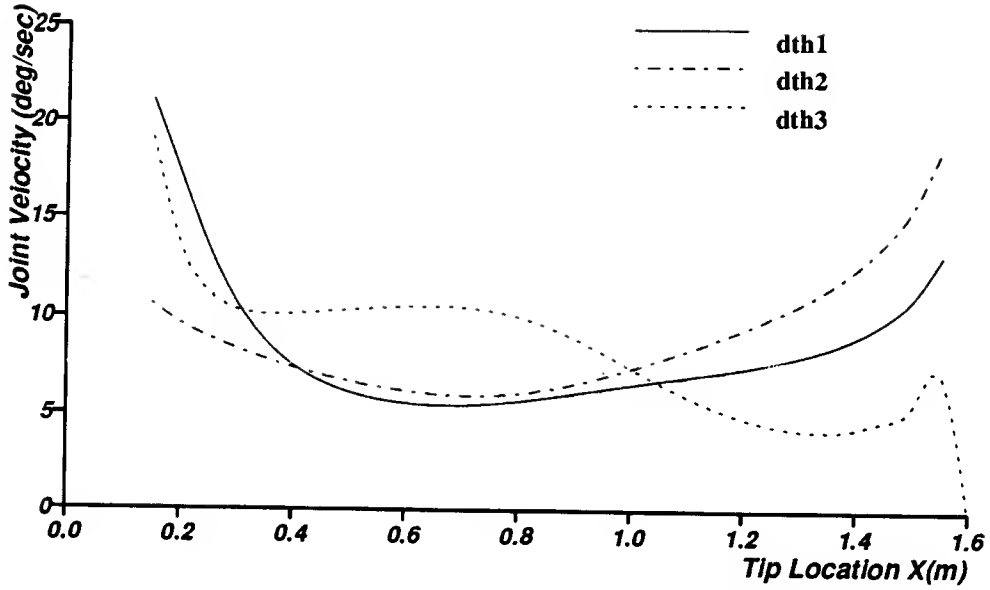


Minors For Configuration A

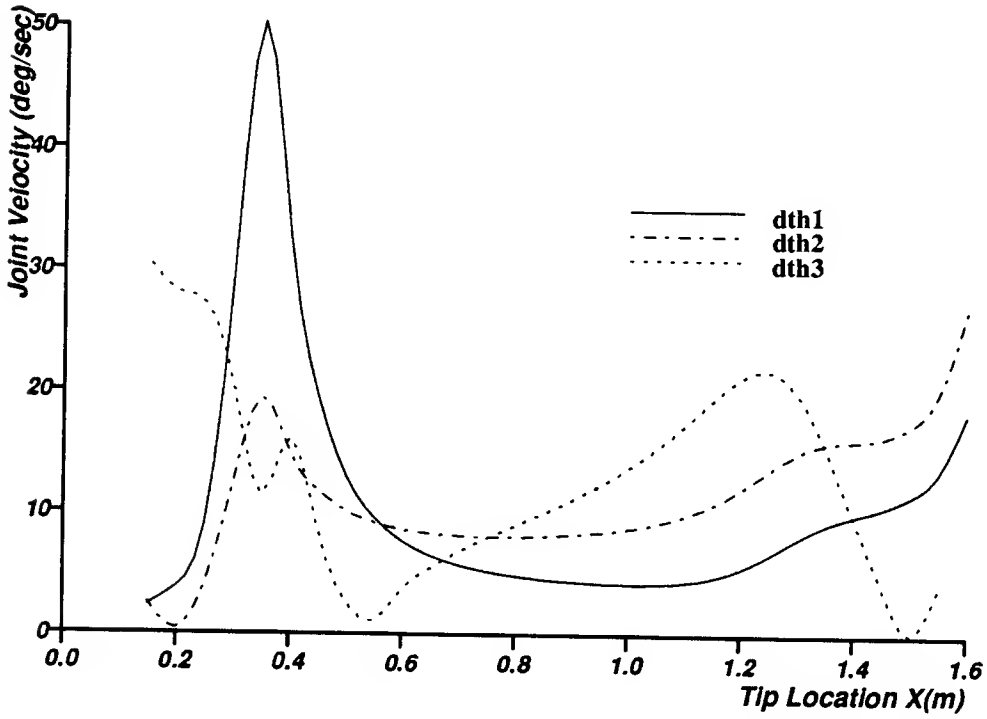


Minors For Configuration B

Figure 6.12: The Minor Values In Configurations A And B, When The New Measure is Used As Performance Measure, Where A And B Refer To The Tip-Motion With Different Initial Configuration.

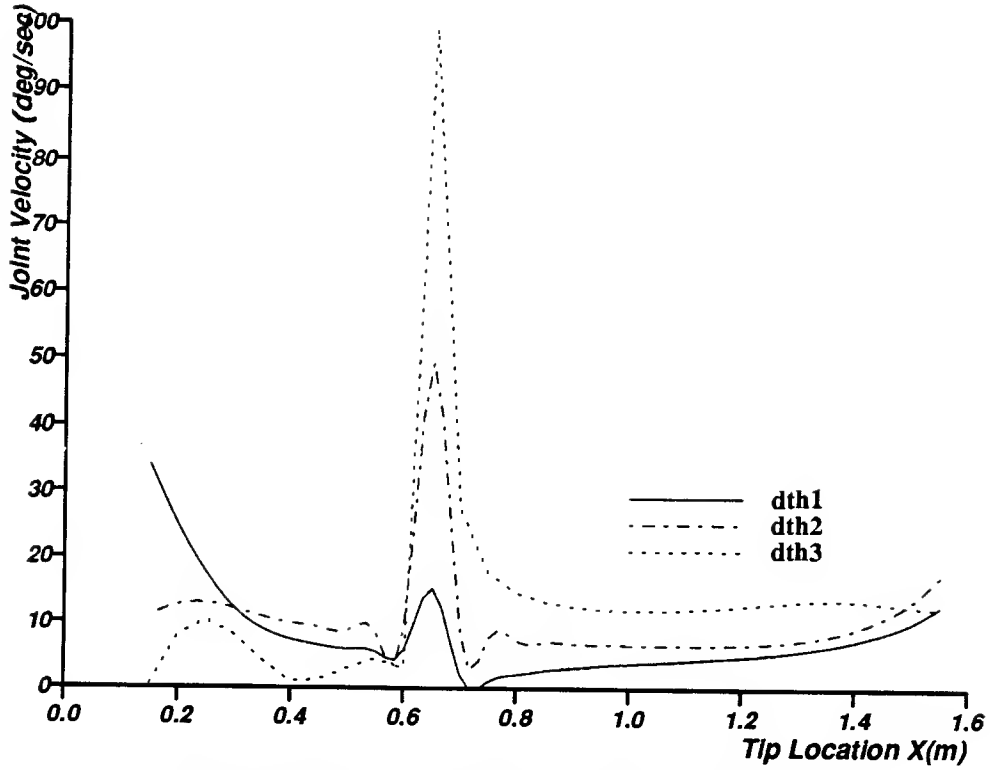


Joint Velocities With The Manipulability: Configuration A

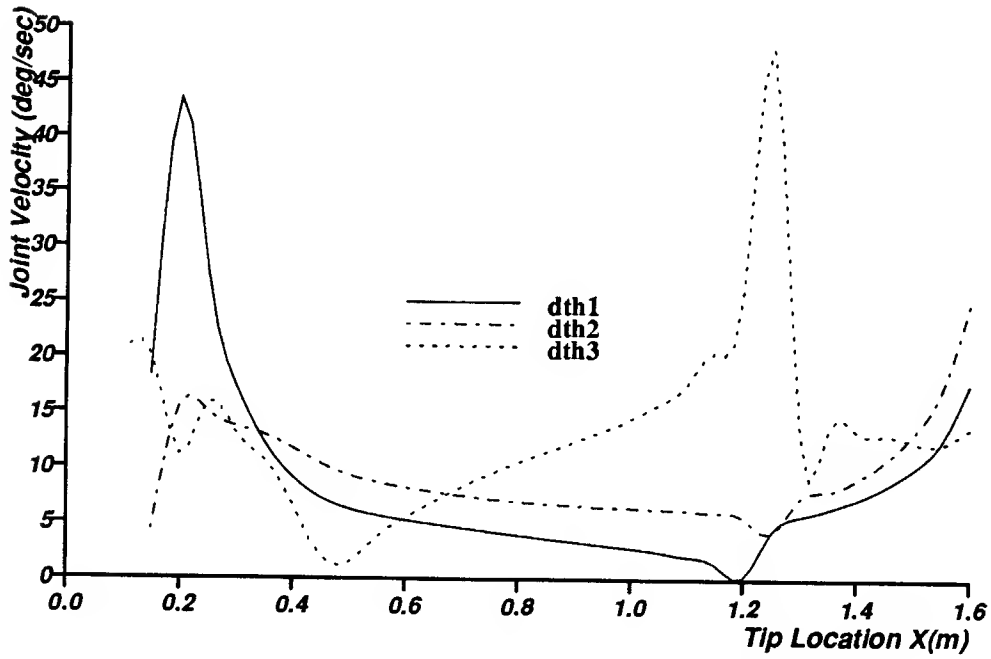


Joint Velocities With The Manipulability: Configuration B

Figure 6.13: The Joint Velocities In Configurations A And B, When The Manipulability is Used As Performance Measure, Where A And B Refer To The Tip-Motion With Different Initial Configurations.

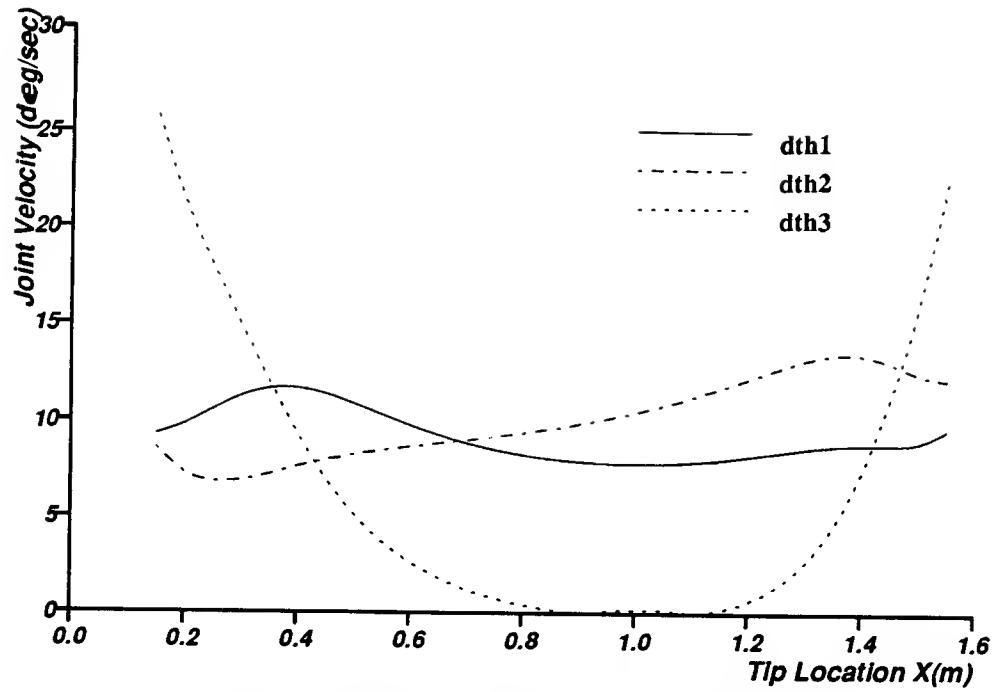


Joint Velocities With The Condition Number: Configuration A

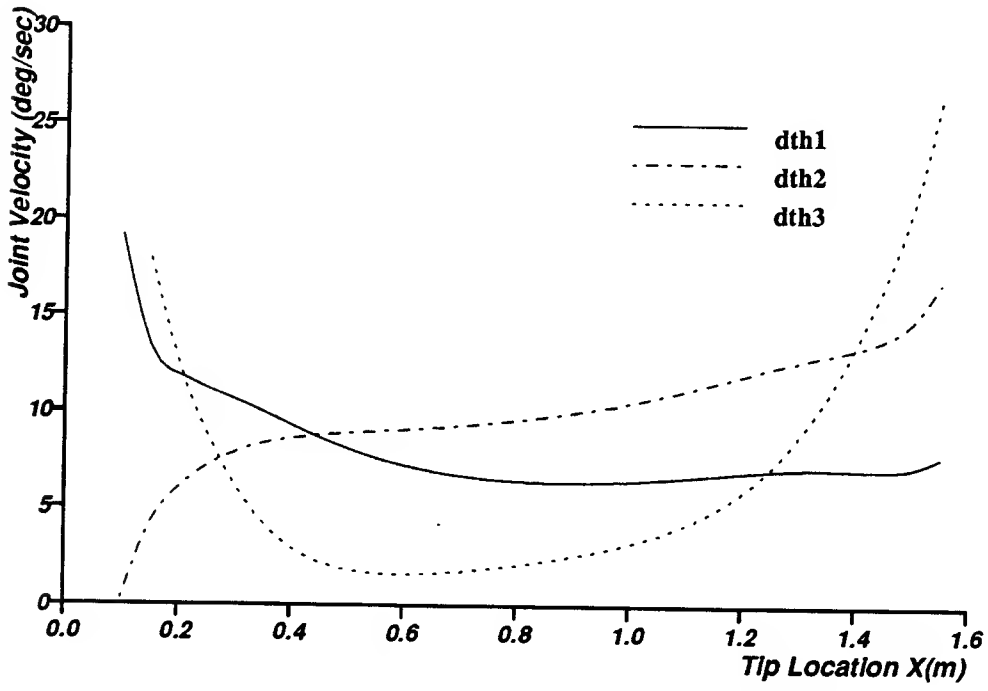


Joint Velocities With The Condition Number: Configuration B

Figure 6.14: The Joint Velocities In Configurations A And B, When The Condition Number is Used As Performance Measure, Where A And B Refer To The Tip-Motion With Different Initial Configurations.



Joint Velocities With The New Measure: Configuration A



Joint Velocities With The New Measure: Configuration B

Figure 6.15: The Joint Velocities In Configurations A And B, When The New Measure is Used As Performance Measure, Where A And B Refer To The Tip-Motion With Different Initial Configurations

Chapter 7

Conclusion

In this thesis, some important aspects of kinematic control problems have been investigated. These aspects may be categorized into two main themes: One is the formulation of a new general the inverse kinematic method for redundant arms. The other is the development of a new dexterity measure.

Finding a concise yet general formulation is important for the exploitation of kinematic redundancy. Furthermore, the fact that the equation is formulated at the *inverse kinematic level* is significant because it has the advantage of directly mapping from workspace to joint space. The relationship between the new formulation and existing ones such as the Resolved Motion Method has been analyzed.

Among various desired performances, the dexterous configurations are not yet well understood. By explicitly defining the dexterity as the *distance* from the kinematic singularity, one can have a good springboard for approaching this rather ambiguous concept. This distance concept in fact helps illuminate the less known fact that for a particular degree of redundancy there are different degrees of distance from singularity. This finding in turn immediately leads to a performance measure for controlling this distance. An analysis of the relationship of the new measure with similar measures confirms that our approach is sound

and useful.

In the light of these two themes, each chapter is reviewed as follows. Chapter 2 covers the derivation of the proposed formula, the interpretation of its characteristics, and estimation of computational efficiency. In addition, the new formula was there compared to the existing formula and it is shown that the new formula is a general, yet concrete expression of the others.

In Chapter 3, we qualitatively compared this formula with the Resolved Motion Method. From the comparison we analytically showed that the proposed formula is in fact a direct mapping counterpart of the other method. That is, the proposed formula was proved to provide directly the equilibrium states at which the joint variables in the other method eventually arrive if the end effector is made to stop at a fixed location. In addition, Resolved Motion Method was shown to lead to non-repeatable behavior. We also proposed a differential relationship, equivalent to the other method, yet based on the proposed formula, which is shown to be slightly more efficient.

In Chapter 4, numerical experiments were made to verify the theoretical and qualitative results so far obtained. Also the numerical efficiency of the new method was compared with the Resolved Motion Method. The theoretical results confirmed through the simulations were as the following:

- the Resolved Motion Method is a linearized equation approximated at the equilibrium points that are exactly determined by the proposed formula;
- the repeatability problem persists in the Resolved Motion Method in transient period as well as at the steady states.

In addition, a practical way to use the proposed formula in a real time control situation was considered in conjunction with an efficient joint interpolation method devised by Taylor. The comparison of efficiency with the other method shows that the proposed formula is more efficient when higher accuracy is required.

In Chapter 5, we have defined the concept of dexterity as a distance from singularity. Then we reviewed the concept of singularity and redundancy for further investigation of the distance concept. We have illustrated that there are different degrees of distance from singularity in the same degree of redundancy, showing that the conventional concept of redundancy is not sufficient to describe this distance. The new distance concept we derived was the number of nonzero minors as well as the magnitude of each minor. On the basis of the new concept, a new performance measure was derived.

In Chapter 6, we have related the new performance measure with the manipulability measure and the condition number. Having investigated the qualitative relationship, we pointed out that the other measures do not have the ability to explicitly prevent minors from becoming zero. Through another series of numerical experiments, the effect of this ability was clearly confirmed. Whereas the two other measures without this ability showed repeatability problem and discontinuous motions, the new measure consistently overcame these problems.

Appendices

Appendix 1: Taylor's Bounded Deviation Algorithm

In this Appendix, we briefly introduce Taylor's bounded deviation algorithm. The algorithm consists of two parts: one part is to generate appropriate numbers of via points and joint angles that guarantee given deviation from Cartesian path; the other part interpolates the joint angles to obtain better resolution. The first part is summarized in the following scheme.

Taylor's Bounded Deviation Joint Paths

1. Compute the joint values θ_0 and θ_1, corresponding to $\mathbf{x}_0$ and $\mathbf{x}_1$, respectively.
2. - Compute the joint space midpoint, $\theta_m = (\theta_1 + \theta_1)/2$, and use it to compute $\mathbf{f}_m = \mathbf{f}(\theta_m)$.
   - Also, compute the workspace midpoint, $\mathbf{x}_m = (\mathbf{x}_0 + \mathbf{x}_1)/2$
3. Compute the deviation between $\mathbf{f}_m$ and $\mathbf{x}_m$, $\delta = |\mathbf{x}_m - \mathbf{f}_m|$
4. If $\delta < \delta^{max}$, we are done. Otherwise, compute the joint value $\theta_x = \mathbf{f}^{-1}(\mathbf{x}_m)$; and apply steps 2—4 *recursively*, for the two segments $(\mathbf{x}_0 \rightarrow \mathbf{x}_m)$ and $(\mathbf{x}_m \rightarrow \mathbf{x}_1)$.

Here, the deviation may be defined as a Euclidean norm of position errors from the desire path. Once the generation of via point is complete, then the joint angles obtained in the process of executing the scheme is now used to generate more intermediate joint angles by interpolation methods. When applying the interpolation methods, if the tip is off the ends of line segments determined by via points, linear interpolation is used. At the transition area from one line segments to another, on the other hand, the quadratic interpolation is used for smooth transitions. This interpolation strategy is implies that the joint velocity is constant at off the transition area, whereas near the transition the joint acceleration made constant.

Appendix 2: The Derivation of The Extended Jacobian Method

In this Appendix, we will prove that (2.10) reduces to the extended Jacobian method, when $n = m + 1$.

The additional equation to resolve the redundancy in the extended Jacobian method is given (Baillieul,1985) in (2.22) and (2.23) as

$$\begin{aligned} G(\boldsymbol{\theta}) &= \mathbf{n}_J \mathbf{h} \\ &= 0 \end{aligned} \tag{A.1}$$

where

$$\begin{aligned} \mathbf{n}_J &= (\triangle_1, \triangle_2, ..., \triangle_n) \\ \triangle_i &= (-1)^{i+1} det(J^1, J^2, ..J^{i-1}, J^{i+1}, .., J^n) \end{aligned}$$

and where J^k is the k-th column vector of the Jacobian matrix, $\mathbf{J}$, derived from (2.1). When $n = m + 1$, our result, (2.10), is specified as

$$\begin{aligned} G_2(\boldsymbol{\theta}) &= [(J^n)^T \mathbf{J_m}^{-1}, -1]\mathbf{h} \\ &= 0 \end{aligned} \tag{A.2}$$

where $(J^n)^T$ is the transpose of the n-th column vector of the Jacobian matrix, $\mathbf{J}$. $\mathbf{J_m}^{-1}$ can be derived as

$$\mathbf{J_m}^{-1} = \frac{1}{D_m} \mathbf{A_m} \tag{A.3}$$

where $\mathbf{A_m}$ is the adjoint matrix of $\mathbf{J_m}$ and D_m is the determinant of $\mathbf{J_m}$. Thus $\mathbf{A_m}$ is expressed as

$$\mathbf{A_m} = \begin{pmatrix} Cof_{11} & Cof_{21} & \dots & Cof_{m1} \\ Cof_{12} & Cof_{22} & \dots & Cof_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ Cof_{1m} & Cof_{2m} & \dots & Cof_{mm} \end{pmatrix} \tag{A.4}$$

where Cof_{ij} is the cofactor of j_{ij} of the Jacobian matrix, $\mathbf{J}$. From (A.2),(A.3), and (A.4), we get

$$(J^n)^T \mathbf{J_m}^{-1} = \frac{1}{D_m}[(J^n)^T A_m^1, (J^n)^T A_m^2, \dots (J^n)^T A_m^m] \tag{A.5}$$

where A_m^i is the i-th column vector of $\mathbf{A_m}$. Since

$$\begin{aligned} (J^n)^T A_m^1 &= j_{1n} Cof_{11} + j_{2n} Cof_{12} + .. + j_{mn} Cof_{1m} \\ &= -\triangle_1 \end{aligned}$$

we get likewise

$$(J^n)^T A_m^i = -\triangle_i \tag{A.6}$$

From the definition of $\triangle_n$, we get

$$\begin{aligned} \triangle_n &= det(\mathbf{J_m}^T) \\ &= det(\mathbf{J_m}) \\ &= D_m \end{aligned} \tag{A.7}$$

Therefore,

$$((J^n)^T \mathbf{J_m}^{-1}, -1) = (-\frac{\triangle_1}{\triangle_n}, -\frac{\triangle_2}{\triangle_n}, \dots, -1) \tag{A.8}$$

Since $\mathbf{J_m}$ is nonsingular — and thus $\triangle_n$ is nonzero, we can multiply by it on both sides of (A.2), resulting in (2.22). Thus we have proved that (2.10) is a general expression which yields the additional equation in the extended Jacobian method.

Appendix 3: The Relationship with Resolved Motion Method At Equilibrium State

From Lemmas 1 and 2, it was shown that column vectors of $\mathbf{Z}^T$ are a set of basis vectors which are orthogonal to $\mathbf{J}$. Thus, row vectors of $\mathbf{J}$, together with column vectors of $\mathbf{Z}^T$ constitute the basis of n-dimensional vector space.

Accordingly, any n-dimensional vector $\mathbf{h}$ can be represented as

$$\mathbf{h} = \mathbf{J}^T\mathbf{h}_1 + \mathbf{Z}^T\mathbf{h}_2 \tag{A.9}$$

where $\mathbf{h}_1$ and $\mathbf{h}_2$ are arbitrary vectors of m and $n - m$ dimensions, respectively. Premultiplying (A.9) by $\mathbf{J}$, we have

$$\mathbf{Jh} = \mathbf{JJ}^T\mathbf{h}_1$$

Thus,

$$\mathbf{h}_1 = (\mathbf{JJ}^T)^{-1}\mathbf{Jh} \tag{A.10}$$

Similarly, if we multiply (A.9) with $\mathbf{Z}$ and solve for $\mathbf{h}_2$, we get

$$\mathbf{h}_2 = (\mathbf{ZZ}^T)^{-1}\mathbf{Zh} \tag{A.11}$$

From (A.9) to (A.11), we obtain the following relationship:

$$(\mathbf{I} - \mathbf{J}^T(\mathbf{JJ}^T)^{-1}\mathbf{J})\mathbf{h} = \mathbf{Z}^T(\mathbf{ZZ}^T)^{-1}\mathbf{Zh}$$

or, from (3.5),

$$(\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{h} = \mathbf{Z}^T(\mathbf{ZZ}^T)^{-1}\mathbf{Zh} \tag{A.12}$$

For a constant location of the end effector (no tip motion), when $\mathbf{dx} = \mathbf{0}$, (3.6), with (A.12), becomes

$$d\boldsymbol{\theta} = \mathbf{Z}^T(\mathbf{ZZ}^T)^{-1}\mathbf{Zh} \tag{A.13}$$

If $\mathbf{Zh} = \mathbf{0}$ as in (2.10), then from (A.13), $d\boldsymbol{\theta} = \mathbf{0}$, which means that joint variables reach an equilibrium state — or a stationary state — which is mostly the optimal configuration for a given tip location. Conversely, if $d\boldsymbol{\theta} = \mathbf{0}$, we have $\mathbf{Zh} = \mathbf{0}$; since the rank of $\mathbf{Z}^T(\mathbf{ZZ}^T)^{-1}$ in (A.13) is $n - m$ and $\mathbf{Zh}$ is an $(n - m)$-dimensional vector.

Therefore, $\mathbf{Zh} = \mathbf{0}$, (2.10), of the proposed method is the necessary and sufficient condition to be satisfied when (3.6) of the RM method reaches its equilibrium state.

Appendix 4: Proof of Theorem 3

In this appendix, we make a proof of Theorem 3. The Jacobian matrix is expressed as the following:

$$\mathbf{J} = \begin{pmatrix} j_{11} & \cdots & j_{1n} \\ & \cdots & \\ j_{m1} & \cdots & j_{mn} \end{pmatrix}$$

Then

$$\mathbf{JJ}^T = \begin{pmatrix} \sum_{k=1}^{n} j_{1k}{}^2 & \cdots & \sum_{k=1}^{n} j_{1k}j_{mk} \\ & \cdots & \\ \sum_{k=1}^{n} j_{mk}j_{1k} & \cdots & \sum_{k=1}^{n} j_{mk}{}^2 \end{pmatrix}$$

In general, the determinant of an $m \times m$ matrix $\mathbf{A}$ is explicitly given as

$$det(\mathbf{A}) = \sum_{\boldsymbol{\sigma}} (a_{1\alpha}a_{2\beta}\cdots a_{m\nu}) det(P_{\boldsymbol{\sigma}}) \tag{A.14}$$

where a_{ij} is the element of $\mathbf{A}$ at i-th row and j-th column, $\boldsymbol{\sigma} = (\alpha, \beta, \cdots, \nu)$ with distinct integers, $\alpha, \beta, \cdots, \nu \quad = 1, 2, \cdots, m$, $P_{\boldsymbol{\sigma}}$ is the permutation matrix, and $\sum_{\boldsymbol{\sigma}}$ means the sum is taken all $m!$ permutations of $\boldsymbol{\sigma}$. Hence, the determinant of $\mathbf{JJ}^T$ is

$$det(\mathbf{JJ}^T) = \sum_{\boldsymbol{\sigma}} (\sum_{k=1}^{n} j_{1k}j_{\alpha k})(\sum_{k=1}^{n} j_{2k}j_{\beta k}) \cdots (\sum_{k=1}^{n} j_{mk}j_{\nu k}) det(P_{\boldsymbol{\sigma}})$$

Expanding this, we have

$$det(\mathbf{JJ}^T) = \sum_{\boldsymbol{\sigma}} (\sum_{k_1, \cdots, k_m = 1}^{n} j_{1k_1}j_{\alpha k_1}j_{2k_2}j_{\beta k_2}, \cdots, j_{mk_m}j_{\nu k_m}) det(P_{\boldsymbol{\sigma}}) \tag{A.15}$$

Here, note that terms that have non-distinct k_i's disappears. For instance, if $k_2 = k_1 = 1$, then $j_{11}j_{\alpha 1}j_{21}j_{\beta 1}, \cdots, j_{mm}j_{\nu m}$ disappears when $\alpha = 1$ and $\beta = 2$. Thus, in Equation A.15, summation is applied only to the terms with distinct k_i's. Note also that the number, p, of different sets of distinct k_i's is

$$p = n\mathrm{C}m.$$

Rearranging Equation A.15, we have

$$det(\mathbf{JJ}^T) = \sum_{k_1, \cdots, k_m = 1}^{n} (\sum_{\boldsymbol{\sigma}} j_{1k}j_{2k}, \cdots, j_{mk} det(P_{\boldsymbol{\sigma}}))(j_{\alpha k}j_{\beta k}, \cdots, j_{mk}j_{\nu k})$$

Here summation $\sum_{k_1,\cdots,k_m=1}^{n}$ may be divided into

$$\sum_{k_1,\cdots,k_m=1}^{n} = \sum_{i=1}^{p}\sum_{\sigma_i}$$

where $\sigma_i = (k_1, k_2, \cdots, k_m)$ is i-th set of p different sets consisting of m distinct k_i's. Therefore Equation A.15 becomes

$$det(\mathbf{J}\mathbf{J}^T) = \sum_{i=1}^{p}\sum_{\sigma_i}(\sum_{\sigma} j_{\alpha k_1} j_{\beta k_2}, \cdots, j_{\nu k_m} det(P_\sigma)) j_{1k_1} j_{2k_2}, \cdots, j_{mk_m} \qquad \text{(A.16)}$$

If we denote a part of Equation A.16 as

$$\Delta_i = \sum_{\sigma} j_{\alpha k_1} j_{\beta k_2}, \cdots, j_{\nu k_m} det(P_\sigma)$$

Comparing this with Equation A.14 shows that Δ_i is the determinant of the transpose of the submatrix made of k_i's column vectors as

$$\Delta_i = det(\lfloor J^{k_1} \;\; J^{k_2} \cdots J^{k_m} \rfloor^T)$$

where J^{k_i} is the k_i-th column vector of the Jacobian matrix. Once a set of k_i's is chosen, the absolute value of Δ_i is fixed; only its sign changes as k_i's make permutations. If we set the absolute value as $|\Delta_i|$, Equation A.16 becomes

$$det(\mathbf{J}\mathbf{J}^T) = \sum_{i=1}^{p}(\pm)\,|\Delta_i|\,(\sum_{\sigma_i} j_{1k_1} j_{2k_2}, \cdots, j_{mk_m} det(P_{\sigma_i}))$$

The facts that the determinant of a matrix is equal to that of transpose of the matrix and that $\mathbf{J}\mathbf{J}^T$ is positive definite imply that

$$det(\mathbf{J}\mathbf{J}^T) = \sum_{i=1}^{p} \Delta_i{}^2.$$

Q.E.D.

Appendix 5: Proof for Derivation of Joint Velocity from the Optimizing Equation

Since $\mathbf{Z}$ spans the null space of j, we have

$$\mathbf{Z}\mathbf{J}^T = \mathbf{0} \tag{A.17}$$

Hence, multiplying both sides of (A.17) by $(\mathbf{J}\mathbf{J}^T)^{-1}$, we have

$$\mathbf{Z}\mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1} = \mathbf{0} \tag{A.18}$$

Using the relationship, $\mathbf{J}^+ = \mathbf{Z}\mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$, we derive from (A.18)

$$\mathbf{Z}\mathbf{J}^+ = \mathbf{0}$$

Meanwhile the definition of the pseudoinverse matrix is again,

$$\mathbf{J}\mathbf{J}^+ = \mathbf{I}_m$$

where $\mathbf{I}_m$ is the identity matrix of rank m. Combining the above two equations results in a composite equation,

$$\begin{bmatrix} \mathbf{J} \\ \mathbf{Z} \end{bmatrix} \mathbf{J}^+ = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \tag{A.19}$$

where the composite matrix is denoted as

$$\mathbf{J}_E = \begin{bmatrix} \mathbf{J} \\ \mathbf{Z} \end{bmatrix} \tag{A.20}$$

Multiplying both sides of (A.19) by $\mathbf{J}_E^{-1}$ results in

$$\mathbf{J}^+ = \mathbf{J}_E^{-1} \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \tag{A.21}$$

Q.E.D.

References

Albala,H, and Angeles,J., 1979, " Numerical Solution to the Input-Output Displacement Equation of General 7R Spatial Mechanism," *Proc. 5th World Cong. Theory Mech. Mechanism*, Montreal, pp. 1008-1011.

Angeles, J., 1985, "On the numerical solution of the inverse kinematic problem," *Int. J. Robotics Research*, 4 no. 2, pp. 21-37.

Baillieul, J., Hollerbach, J.M., and Brockett, R., 1984, "Programming and control of kinematically redundant manipulators," *Proc. 23rd IEEE Conference on Decision and Control*, Las Vegas, Nevada, December 12-14, pp. 768-774.

Baillieul, J., 1985, "Kinematic programming alternatives for redundant manipulators," *Proc. IEEE Conf. Robotics and Automation*, St. Louis, Mar. 25-28, pp. 722-728.

Baillieul, J., 1986, "Avoiding obstacles and resolving kinematic redundancy," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp. 1698-1704.

Benati, M., Morasso, P., and Tagliasco, V., 1982, "The inverse kinematic problem for anthropomorphic manipulator arms," *ASME J. Dynamic Systems, Meas., Control*, 104, pp. 110-113.

Benhabib, B., Fenton, R.G., and Goldenberg, A.A., 1984, "Point-to-point control of seven degrees of freedom robots for assembly operations," *Proc. 3rd Canadian CAD/CAM and Robotics Conf.*, Toronto, pp. 10/1-10/7.

Benhabib, B., Goldenberg, A.A., and Fenton, R.G., 1985, "A solution to the

inverse kinematics of redundant manipulators," *Proc. American Control Conf.*, Boston, June 19-21, pp. 368-374.

Ben-Israel,A. and Greville,T., 1980, *Generalized Inverses: Theory and Applications*, New York, Robert E. Krieger Publishing Co..

Borrel, P., and Liegeois, A., 1986, "A study of multiple manipulator inverse kinematic solutions with applications to trajectory planning and workspace determination," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp. 1180-1185.

Chang, P.H., 1986, "A closed-form solution for the control of manipulators with kinematic redundancy," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp. 9-14.

Espiau, B., and Boulic, R., 1985, "Collision avoidance for redundant robots with proximity sensors," *Preprints 3rd International Symposium of Robotics Research*, Gouvieux (Chantilly), France, October 7-11, pp. 94-102.

Featherstone, R., 1983, "Position and velocity transformations between robot end-effector coordinates and joint angles," *Int. J. Robotics Research*, 2 no. 2, pp. 35-45.

Goldenberg, A., Benhabib, B., and Fenton, R.G., 1985, "A complete generalized solution to the inverse kinematics of robots," *IEEE J. Robotics and Automation*, 1, pp. 14-20.

Goldenberg, A.A., and Lawrence, D.L., 1985, "A generalized solution to the inverse kinematics of robotic manipulators," *ASME J. Dynamic Systems, Meas., Control*, 107, pp. 103-106.

Gupta, K.C., and Kazerounian, K., 1985, "Improved numerical solutions of inverse kinematics of robots," *Proc. IEEE Int. Conf. Robotics and Automation*, St. Louis, March 25-28, pp. 743-748.

Hamming,R., 1973, *Numerical Methods for Scientists and Engineers (2nd ed.)*, New York, McGraw-Hill, Inc..

Hartenburg, R. S., and Denavit, J., 1964, *Kinematic Synthesis of Linkages*, New York, McGraw-Hill.

Hollerbach,J.,Suh,K., 1986, "Redundant Resolution of Manipulators Through Torque optimization," *A.I.Memo 882*, Cambridge, January.

Hollerbach, J.M., 1985, "Optimum kinematic design for a seven degree of freedom manipulator," *Robotics Research: The Second International Symposium*, edited by H. Hanafusa and H. Inoue, Cambridge, Mass., MIT Press, pp. 215-222.

Klein, C.A., 1985, "Use of redundancy in the design of robotic systems," *Robotics Research: The Second International Symposium*, edited by H. Hanafusa and H. Inoue, Cambridge, Mass., MIT Press, pp. 207-214.

Klein, C.A., and Huang, C.H., 1983, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Systems, Man, Cybern.*, SMC-13, pp. 245-250.

Konstantinov, M.S., Markov, M.D., and Nenchev, D.N., 1981, "Kinematic control of redundant manipulators," *Proc. 11th Int. Symp. on Industrial Robots*, Tokyo, Japan, October 7-9, pp. 561-568.

Liegeois, A., 1977, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Systems, Man, Cybern.*, SMC-7, pp. 868-871.

Luh, J.Y.S., and Gu, Y.L., 1985, "Industrial robots with seven joints," *Proc. IEEE Conf. Robotics and Automation*, St. Louis, Mar. 25-28, pp. 1010-1015.

Maciejewski, A.A., and Klein, C.A., 1985, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robotics Research*, 4 no. 3, pp. 109-117.

Mathlab Group, 1983, "MACSYMA Reference Manual," Version 10, Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge, MA..

Moré,J., Garbow,B., and Hillstrom,K., 1980, *User Guide for MINPACK-1*, Argonne National Laboratory.

Nakamura, Y., 1985a, Kinematical Studies on the Trajectory Control of Robot Manipulators, Ph.D. Thesis, Kyoto Univ., June.

Nakamura, Y., and Hanafusa, H., 1985b, "Task priority based redundancy control of robot manipulators," *Robotics Research: The Second International Symposium*, edited by H. Hanafusa and H. Inoue, Cambridge, Mass., MIT Press, pp. 155-162.

Nakamura, Y., and Hanafusa, H., 1985c, "Inverse kinematic solution with singularity robustness for robot manipulator control," *ASME Winter Annual Meeting:*

Robotics and Manufacturing Automation, PED-Vol. 15, Miami Beach, Nov. 17-22, pp. 193-204.

Oh,S.Y., Orin,D.E., and Bach,M, 1984, "An Inverse Kinematic Solution for Kinematically Redundant Robot Manipulators," *Journal of Robot Systems*, Summer, pp. 235-249.

Orin, D.E., and Schrader, W.W., 1984, "Efficient Jacobian determinaton for robot manipulators," *Robotics Research: The First International Symposium*, edited by M. Brady and R. Paul, Cambridge, Mass., MIT Press, pp. 727-734.

Paul, R.P., September, 1975, "Manipulator Path Control," *Proc. Intl. Conf. Cybernetics and Society*, New York, pp. 147-152.

Paul, R.P., 1979, "Manipulator Cartesian path control," *IEEE Trans. Systems, Man, Cybernetics*, SMC-9, pp. 702-711.

Paul, R.P., 1981, *Robot Manipulators: Mathematics, Programming, and Control*, Cambridge, Mass, MIT Press.

Paul, R. P., and Stevenson, C. N., 1983, "Kinematics of robot wrists," *Int. J. Robotics Research*, 2 no. 1, pp. 31-38.

Pieper, D.L., 1968, The Kinematics of Manipulators under Computer Control, Ph.D. Thesis, Stanford University, Computer Science.

Rabinowitz, P., 1970, *Numerical Methods for Nonlinear Algebraic Equations*, New York, Gordon and Breach Science Publisher, Inc..

Salisbury, J. K., and Craig, J. J., 1982, "Articulated Hands: Force Control and Kinematic Issues," *The International Journal Of Robotics Research*, in press.

Strang, G., 1980, *Linear Algebra and its applications*, New York, Academic Press.

Taylor, R.H., 1979, "Planning and execution of straight-line manipulator trajectories," *IBM Journal of Research and Development*, 23, pp. 424-436.

Uchiyama, M., 1979, "Study on Dynamic Control of Artificial Arm - Part 1," *JSME*, in press.

Uchiyama, M., Shimizu, K., and Hakomori, K., 1985, "Performance evaluation of manipulators using the Jacobian and its application to trajectory planning," *Robotics Research: The Second International Symposium*, edited by H. Hanafusa

and H. Inoue, Cambridge, Mass., MIT Press, pp. 447-456.

Whitney, D.E., 1969, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Machine Systems*, MMS-10, pp. 47-53.

Whitney, D.E., 1972, "The mathematics of coordinated control of prosthetic arms and manipulators," *ASME J. Dynamic Systems, Meas., Control*, pp. 303-309.

Yoshikawa, T., 1984, "Analysis and Control of Robot Manipulators with Redundancy," *Robotics Research: The First International Symposium*, edited by M. Brady and R. Paul, Cambridge, Mass., MIT Press, pp. 439-446.

Yoshikawa, T., 1985a, "Manipulatability of Robotic Mechanisms," *Robotics Research: The Second International Symposium*, edited by H. Hanafusa and H. Inoue, Cambridge, Mass., MIT Press, pp. 439-446.

Yoshikawa, T., 1985b, "Analysis and Design of Articulated Robot Arms from the Viewpoint of Dynamic Manipulability," *Preprints 3rd International Symposium of Robotics Research*, Gouvieux(Chantilly), France, October 7-11, pp. 150-156.

List of Figures

List of Tables

CS-TR Scanning Project
Document Control Form

Date : 9/14/95

Report # AI-TR-1022

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☒ Technical Report (TR) ☐ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 130(137-IMAGES)
Not to include DOD forms, printer instructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:_____

Check each if included with document:

☒ DOD Form (2) ☐ Funding Agent Form ☒ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages(by page number):_____

Photographs/Tonal Material (by page number):_____

Other (note description/page number):

Description :	Page Number:
IMAGE MAP:	(1-2)
UN#TITLE AND ABSTRACT PAGES	(3-130)
PAGES#'ED 2-5, NO PAGE 6, 7-130	(131-137)
SCANCONTROL, COVER, DOD(2), TRGTS(3)	

Scanning Agent Signoff:

Date Received: 9/14/95 Date Scanned: 9/18/95 Date Returned: 9/21/95

Scanning Agent Signature: Michael W. Cook

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AI-TR 1022
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): Analysis and Control of Robot Manipulators with Kinematic Redundancy
5. TYPE OF REPORT & PERIOD COVERED: technical report
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): Pyung Hun Chang
8. CONTRACT OR GRANT NUMBER(s): N00014-80-C-0505, N00014-86-K-0685

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS:
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. REPORT DATE: February 1988
13. NUMBER OF PAGES: 132

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office):
Office of Naval Research
Information Systems
Arlington, VA 22217

15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report):
Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES:
None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
kinematic control
redundant manipulators

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

A closed-form solution formula for the kinematic control of manipulators with redundancy is derived, using the Lagrangian multiplier method. Differential relationship equivalent to the Resolved Motion Method has been also derived. The proposed method is proved to provide with the exact equilibrium state for the Resolved Motion Method. This exactness in the proposed method fixes the repeatability problem in the Resolved Motion Method, and establishes a fixed

Block 20 cont.

transformation from workspace to the joint space. Also the method, owing to the exactness, is demonstrated to give more accurate trajectories than the Resolved Motion Method.

In addition, a new performance measure for redundancy control has been developed. This measure, if used with kinematic control methods, helps achieve dexterous movements including singularity avoidance. Compared to other measures such as the manipulability measure and the condition number, this measure tends to give superior performances in terms of preserving the repeatability property and providing with smoother joint velocity trajectories.

Using the fixed transformation property, Taylor's *Bounded Deviation Paths* Algorithm has been extended to the redundant manipulators.

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**